posDesigner

User Manual

symbol®
The Enterprise Mobility Company™

posDesigner
User Manual

72E-64641-02
Revision A
July 2004

© 2001-2004 by Symbol Technologies, Inc. All rights reserved.

No part of this publication may be reproduced or used in any form, or by any electrical or mechanical means, without permission in writing from Symbol. This includes electronic or mechanical means, such as photocopying, recording, or information storage and retrieval systems. The material in this manual is subject to change without notice.

The software is provided strictly on an "as is" basis. All software, including firmware, furnished to the user is on a licensed basis. Symbol grants to the user a non-transferable and non-exclusive license to use each software or firmware program delivered hereunder (licensed program). Except as noted below, such license may not be assigned, sublicensed, or otherwise transferred by the user without prior written consent of Symbol. No right to copy a licensed program in whole or in part is granted, except as permitted under copyright law. The user shall not modify, merge, or incorporate any form or portion of a licensed program with other program material, create a derivative work from a licensed program, or use a licensed program in a network without written permission from Symbol. The user agrees to maintain Symbol's copyright notice on the licensed programs delivered hereunder, and to include the same on any authorized copies it makes, in whole or in part. The user agrees not to decompile, disassemble, decode, or reverse engineer any licensed program delivered to the user or any portion thereof.

Symbol reserves the right to make changes to any software or product to improve reliability, function, or design.

Symbol does not assume any product liability arising out of, or in connection with, the application or use of any product, circuit, or application described herein.

No license is granted, either expressly or by implication, estoppel, or otherwise under any Symbol Technologies, Inc., intellectual property rights. An implied license only exists for equipment, circuits, and subsystems contained in Symbol products.

Symbol, Spectrum One, and Spectrum24 are registered trademarks of Symbol Technologies, Inc. Other product names mentioned in this manual may be trademarks or registered trademarks of their respective companies and are hereby acknowledged.

Symbol Technologies, Inc.
One Symbol Plaza
Holtsville, New York 11742-1300
http://www.symbol.com

Contents

Chapter 1. posDesigner Overview

Chapter 2. Supported Graphics and Image Formats

Colors, Resolution, and Dithering 2-2
Importing Graphic Images to the TCE 2-3
Available Point of Sale Fonts 2-5

Chapter 3. Using posDesigner to Create New Screens

Chapter 4. posDesigner Screen Templates

scrWelcome 4-2
scrLineItems 4-3
scrSelectPayment 4-4
scrMsrCredit 4-5
scrMsrDebit 4-7

scrSignature 4-8
scrPin. 4-11
scrCashBack 4-13
scrThankYou 4-14
scrCustom 4-15

Chapter 5. Toolbars, Screens, and Controls

Control Property OK to All Option 5-1
Control Property Cancel All Option. 5-2
Controls Toolbar 5-2
    Creating a New MSR Control. 5-4
Grid Toolbar 5-5
Alignment Toolbar 5-5
Main Toolbar. 5-6
Navigation Toolbar 5-8
Screen Toolbar 5-8
Screen Properties 5-9
Screens Pane 5-10
Controls Pane 5-11
Background Image Control 5-12
Image List Control. 5-13
Button Control. 5-14
Check Box Control. 5-16
Radio Button Control. 5-17
Line Display Control 5-18
MSR Control 5-20
Signature Control 5-22
PIN Control 5-24
Static Text Label Control 5-27
Variable Text Label Control 5-28
Input Field Control. 5-29
Installation File Block (IFB) 5-30
Running a Script on the POS Terminal 5-31

Chapter 6. posDesigner Styles

Style Example 6-2

Chapter 7. Project INI Files

Chapter 8. posDesigner Actions

ActionDelay 8-3
BeginTransaction 8-3
ClearTransaction 8-3
ControlVisibilitySet 8-4
DeviceReset 8-4
DownloadMode 8-4
EndTransaction 8-4
FireEvent 8-4
GotoScreen 8-5
GotoScreenVariable 8-5
InputFieldAppend 8-5
InputFieldAppendVariable 8-6
InputFieldClear 8-6
InputFieldSave 8-6
LineDisplayClear 8-6
LineDisplayDataAdd 8-6
LineDisplayRefresh 8-6
LineDisplayScrollDown 8-7
LineDisplayScrollUp 8-7
PlaySound 8-7
PlaySystemSound 8-8
SignatureClear 8-8
SignatureEndCapture 8-8
TextLabelAppend 8-8
TextLabelSave 8-8
TextLabelSet 8-9
TouchPadOFF 8-9
TouchPadON 8-9
VariableClear 8-9

VariableCopy 8-9
VariableSet 8-9

About This Guide

Introduction

The *posDesigner User Manual* provides general instructions for using posDesigner, a real-time graphic screen flow editor used for creating POS (Point Of Sale) applications running on a Symbol POS interactive terminal (PD family of products and iPOS TCE).

Chapter Descriptions

Topics covered in this guide are as follows:

- *Chapter 1, posDesigner Overview*, provides a basic overview of the application.
- *Chapter 2, Supported Graphics and Image Formats*, lists supported graphic and image formats, as well as available colors and resolutions; it also contains instructions for importing images to the TCE.
- *Chapter 3, Using posDesigner to Create New Screens*, explains how to create new screens.

- *Chapter 4, posDesigner Screen Templates*, explains how to design screens, and how to add controls to that enable users to perform such tasks as selecting a payment type, swiping a credit or debit card, providing a signature, and so on.
- *Chapter 5, Toolbars, Screens, and Controls*, explains posDesigner toolbars, which enable developers to navigate the screens used to build point-of sale screen flows; screenand control panes, which enable developers to jump from one screen to another and select a control for editing; and controls, which enable developers to define the look and behavior of the point-of-sale screen flow.
- *Chapter 6, posDesigner Styles*, explains how to create styles (collections of images used to replace existing images in a script) to apply to preexisting scripts.
- *Chapter 7, Project INI Files*, provides information on creating or uploading INI files (text files that contain varilable and value pairs) to set new default values for preexisting scripts.
- *Chapter 8, posDesigner Actions*, explains the actions caused by events that enact specific controls. A few of these actions (such as **GotoScreen**) are simulated in the emulator; most are not. Actions enable the developer to set variables, jump from one screen to the next and save signature data.

Notational Conventions

The following conventions are used in this document:

- Italics are used to highlight specific items in the general text, and to identify chapters and sections in this and related documents.
- Bullets (•) indicate:
  - action items
  - lists of alternatives
  - lists of required steps that are not necessarily sequential
- Sequential lists (those that describe step-by-step procedures) appear as numbered lists.

Service Information

If you have a problem with your equipment, contact the Support Center for your region. Before calling, have the model number, serial number, and several of your bar code symbols at hand.

Call the Support Center from a phone near the scanning equipment so that the service person can try to talk you through your problem. If the equipment is found to be working properly and the problem is symbol readability, the Support Center will request samples of your bar codes for analysis at our plant.

If your problem cannot be solved over the phone, you may need to return your equipment for servicing. If that is necessary, you will be given specific directions.

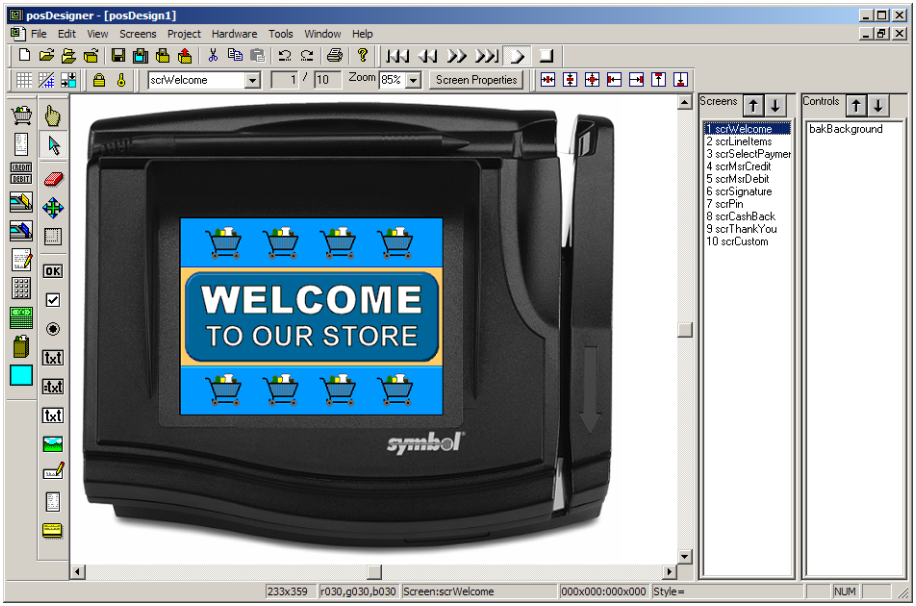
Symbol Technologies is not responsible for any damages incurred during shipment if the approved shipping container is not used. Shipping the units improperly can possibly void the warranty. If the original shipping container was not kept, contact Symbol to have another sent to you.

1

posDesigner Overview

posDesigner is a real-time graphic screen flow editor designed for creating POS (Point Of Sale) applications running on a Symbol POS interactive terminal (PD family of products and iPOS TCE). posDesigner includes a POS terminal emulator simulating POS actions such as button presses, animations, and screen flow movements. The posDesigner User Manual supports posDesigner version 6.

The posDesigner interface enables the developer to design and view a screen flow before uploading it to a Symbol POS interactive terminal. The developer is able to view the device screen flow without using a terminal. Once the POS application is completed, it can be uploaded to the terminal by clicking a button.



posDesigner includes standardized POS screens to design POS screen flows. The posDesigner editor also supports the creation of custom screens. The posDesigner drag and drop graphic interface enables (often with the single click of a button) new screens to be added to the screen flow.

2 Supported Graphics and Image Formats

posDesigner supports standard graphic formats (JPEG, GIF, BMP) and their variations as well as numerous signature formats. Graphic images supplied by the developer can be used for creating customized buttons, backgrounds, and ads.

The graphic (raster bitmapped) image formats supported are listed in the table below.

Format	*1-bit*	*4-bit*	*8-bit*	*24-bit*
BMP	2 colors	16 colors	256 colors	16,777,216 colors
CUR	2 colors	16 colors	256 colors	N/A
DIB	2 colors	16 colors	256 colors	16,777,216 colors
EMF	2 colors	16 colors	256 colors	16,777,216 colors
GIF	2 colors	16 colors	256 colors	N/A
ICO	2 colors	16 colors	256 colors	N/A

Format	*1-bit*	*4-bit*	*8-bit*	*24-bit*
JPEG	N/A	N/A	256 b/w shades	16,777,216 colors
PNG	2 colors	16 colors	256 colors	N/A
TIF	2 colors	16 colors	256 colors	16,777,216 colors
WMF	2 colors	16 colors	256 colors	16,777,216 colors

The table below lists supported signature (vector line) image formats.

Format	Compression Type	Compression Level	Image Quality	Average Compression Ratio	Average Size in Bytes
FBP	none	N/A	excellent	1:1	2000
SIG	lossless	N/A	excellent	5:4	1600
VBC	lossless	0	excellent	4:1	500
VBC	lossy	5	good	5:1	400
NLC	lossy	N/A	good	5:1	400
CMP	lossy	N/A	poor	10:1	200
VBC	lossy	10	poor	11:1	180
VBC	lossy	14	good	16:1	125

For more information on the signature formats supported, refer to the *posBuilder Software Development Toolkit* (p/n 72E-63755-01).

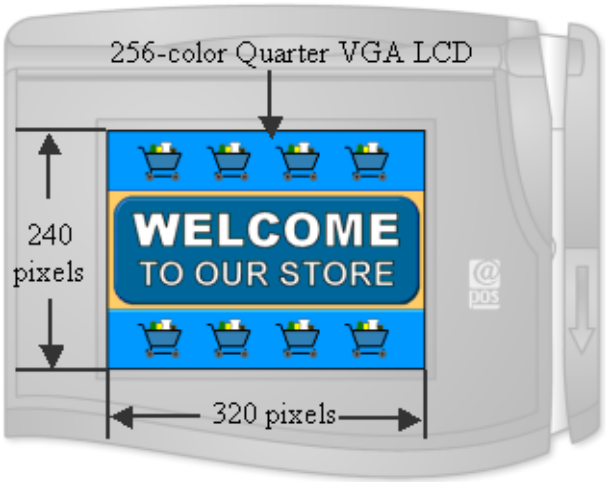
2.1 Colors, Resolution, and Dithering

The Symbol TCE POS terminal contains a 256-color quarter VGA (320x240 pixel) LCD. The 256 colors available on the Symbol TCE cannot be changed. The 256-color palette used is the WST (Windows Web-Safe Skin Tone) palette. The WST is a 256-color optimized palette supporting all 216 standard web safe colors and 20 standard Microsoft Windows system colors. The WST is supplemented with a 28-color international skin tone palette. 8 of the 20 Microsoft Windows system colors are present in the 216 color web-safe palette. Duplicate colors are omitted from the WST palette to allow the complete 28-color international skin tone palette.

The Symbol PD8500 POS terminal contains a 4096-color quarter VGA (320x240 pixel) LCD.

The Symbol DP4500 POS terminal contains a monochrome blue tinted 256-shade quarter VGA (320x240 pixel) LCD.

The posDesigner emulator view can simulate all three hardware modes.



2.2 Importing Graphic Images to the TCE

This chapter only applies to the TCE. When importing images to the PD8500 or PD4500, images remain untouched.

When importing images to the TCE POS terminal, images imported into posDesigner remain untouched if matching the requirements of the TCE POS terminal display specifications. If they do not match the requirements of the target terminal, they are pre-formatted to match. If an image contains colors other than those found in the WST palette, the image is pre-formatted when imported.

When imported, images require pre-formatting by posDesigner. Images can be dithered if dithering is selected. Four dithering options are available: None, Low, Medium, and High.

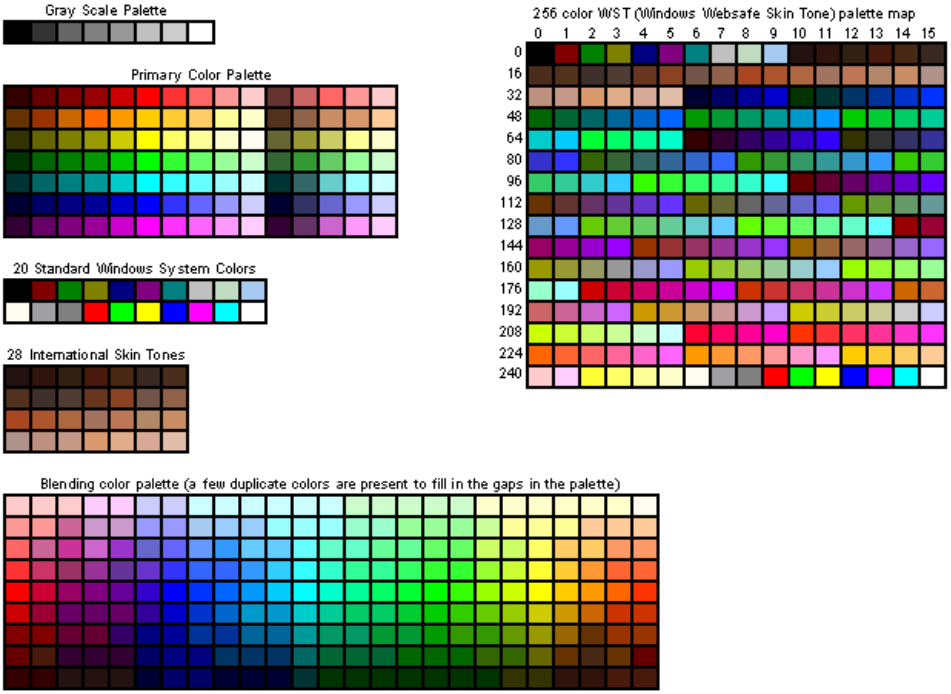
Dithering simulates colors not available. It uses various patterns of existing colors to simulate new colors. For example, a pattern of red and blue might be used to simulate the color purple. Dither color patterns can be very effective for CRT (cathode-ray tube) displays where the colored pixels in the dither patterns tend to blend into each other to create the appearance of a new color. Dither color patterns are not that effective on LCD displays. On LCD displays, the pixels appear more distinct and

tend not to blend into each other, producing a very grainy effect. If you are designing your screens using a standard CRT display, don't expect the dithered images to look as nice as on an LCD display.

The **None** option bypasses dithering completely. The colors in the imported image not matching those in the WST palette are replaced with similar colors in the WST palette in order to best represent the intended image. **Low** produces the least amount of dithering. **Medium** produces a moderate amount of dithering. **High** produces the highest level of dithering. Each dithering level could potentially produce the best result depending on the specific image being imported.

An image displaying people, plants or other natural imagery looks best with the dither option set to High. Man made images of text and straight lines look best using the Low or None dither settings.

An image using colors only found in the 256-color WST palette (such as an image created using the 20-color Windows system palette or the 216-color web-safe palette) is never dithered no matter which dithering option is used. Dithering is only performed when it is needed. It is only used on portions of images containing colors not found in the 256-color WST palette.



2.3 Available Point of Sale Fonts

The PD8500 (system version 2.5 running posPortalCE version 6.5) currently supports 4 fonts:

Font	***Pitch***	***Sample Font Characters***
Arial	proportional	ABCDEFGHIJKLMNOPQRSTUVWXYZ abcdefghijklmnopqrstuvwxyz 1234567890
Courier New	fixed-width	ABCDEFGHIJKLMNOPQRSTUVWXYZ abcdefghijklmnopqrstuvwxyz 1234567890
Tahoma	proportional	ABCDEFGHIJKLMNOPQRSTUVWXYZ abcdefghijklmnopqrstuvwxyz 1234567890
Times New Roman	proportional	ABCDEFGHIJKLMNOPQRSTUVWXYZ abcdefghijklmnopqrstuvwxyz 1234567890

The iPOS TCE only supports the `Courier New` font. The fonts used in the hardware are identical to the fonts bundled with the Microsoft Windows operating system.

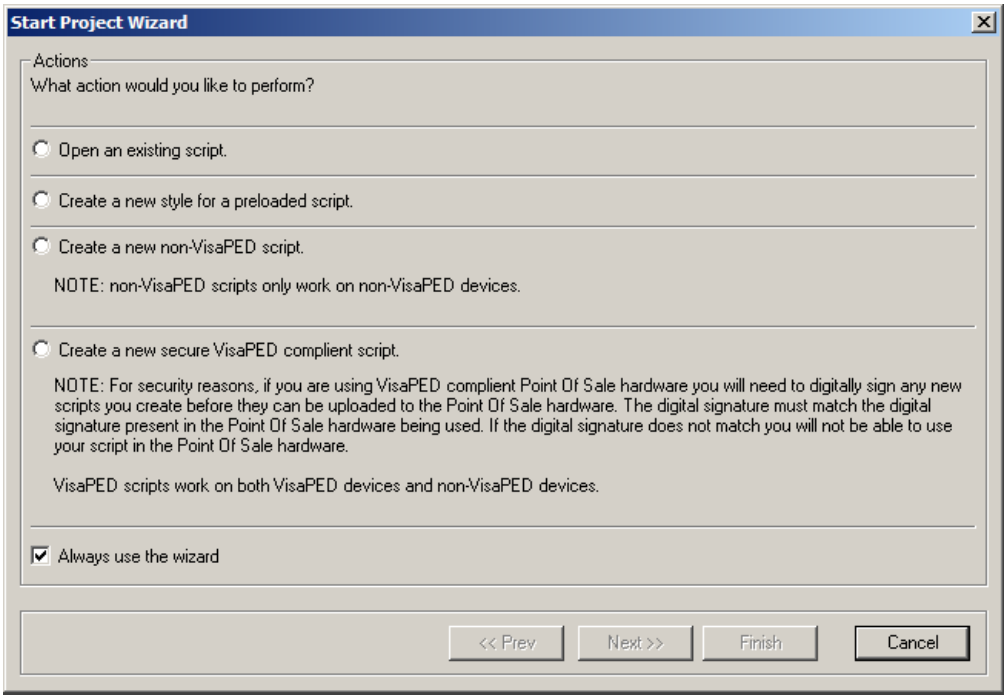
The `Courier New` font is the only font available that has a fixed-width pitch (each character in the font has the same width). All of the other fonts have a proportional pitch (each character has its own width).

The `Courier New` font is typically used when characters are horizontally aligned. The Line Display Control uses Courier New font.

3

Using posDesigner to Create New Screens

The first time you use posDesigner the Start Project Wizard is displayed.


Start Project Wizard
Actions
What action would you like to perform?
Open an existing script.
Create a new style for a preloaded script.
Create a new non-VisaPED script.
NOTE: non-VisaPED scripts only work on non-VisaPED devices.
Create a new secure VisaPED complient script.
NOTE: For security reasons, if you are using VisaPED complient Point Of Sale hardware you will need to digitally sign any new scripts you create before they can be uploaded to the Point Of Sale hardware. The digital signature must match the digital signature present in the Point Of Sale hardware being used. If the digital signature does not match you will not be able to use your script in the Point Of Sale hardware.
VisaPED scripts work on both VisaPED devices and non-VisaPED devices.
Always use the wizard
<< Prev
Next >>
Finish
Cancel

The project wizard prompts you with 4 choices:

1. Open an existing script.
2. Create a new style for a preloaded script.
3. Create a new non-VisaPED script.
4. Create a new secure VisaPED complient script.

To create a new document select either option 3 or 4. Select 4 if the target POS terminal is VisaPED compliant, otherwise Select 3. When done press **Next**.

Next a **New** file dialog appears asking for the type of project to create. There are two choices:

1. posDesigner Interactive Script (*.dis)
2. posIFB Interactive Script (*.ifb)

Choose option 1 for most cases. Choosing option 1 creates a standard project. Choosing option 2 creates a packaged project with all project files packaged together as one file; option 2 is slower. When done press **OK**.

Next the **New Project** dialog box displays. Follow these steps:

1. Enter the file name of the new project in the **Project Name** field.
2. Enter the location (path) of the file in the **Location** field.
3. Press **OK** to create the new posDesigner project.

   The new project displays blank with no screens.
4. Click one of the buttons within the **New Screen** tool bar to create a screen:

scrWelcome	Creates a welcome screen.
scrLineItems	Creates a line item display screen.
scrSelectPayment	Creates a select payment menu screen.
scrMsrCredit	Creates a MSR credit card swipe screen.
scrMsrDebit	Creates a MSR debit card swipe screen.
scrSignature	Creates a signature entry screen.
scrPin	Creates a PIN entry screen.
scrCashBack	Creates a cash back menu screen.
scrThankYou	Creates a thank you screen.
scrBlank	Creates a blank screen.

4

posDesigner Screen Templates

Customize screen templates by manipulating posDesigner controls or adding new buttons and backgrounds images.

There are several additional controls (Signature, PIN, and MSR) that cannot be added to a screen. To create a screen with one of these controls, the developer is required to use a screen template containing the control.

When adding important text in the screen flow, ensure a high contrast text is used. Black text on a white background is a Symbol recommended easy to read high contrast color combination Red text on a blue background is a difficult to read low contrast color combination difficult for color-blind people to read. Some customers are vision impaired. Symbol recommends making screen text large enough for people with bad eyesight to read.

When designing screens, Symbol recommends keeping the screen flow consistent with other applications used your retail environment.

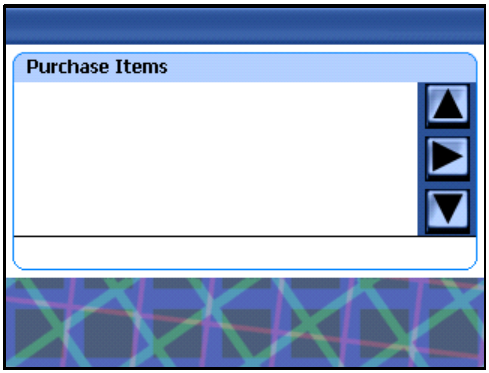
4.1 scrWelcome

The scrWelcome screen contains just the Background object "**bakBackground**".

By default, the scrWelcome screen displays until the ECR (Electronic Cash Register) issues a command to display the next screen in the transaction screen flow.

Set the **scrWelcome** screen to move to the next screen (typically the **scrLineItems** screen) by adding a **GotoScreen** action for the On touch property of the background image, or by adding a **Next** button to the screen with a **GotoScreen** action for its On button pressed property.

4.2 scrLineItems



The scrLineItems screen contains eight objects:

- Background: **bakBackground**
- Image List: **imgHeaderBar**
- Line Display: **lstLineItems**
- Variable Text Label: **txtMsg**
- Image List: **imgAd**
- Button: **btnScrollUp**
- Button: **btnNext**
- Button: **btnScrollDown**

Use the scrLineItems screen to display ads and purchased items as they are scanned.

Use the **imgHeaderBar** control to display a logo.

The **lstLineItems** control is used to display purchased items. The contents of the Line Display object are controlled by the host ECR (Electronic Cash Register).

Use the **txtMsg** control to display text (such as a running total) sent from the host ECR.

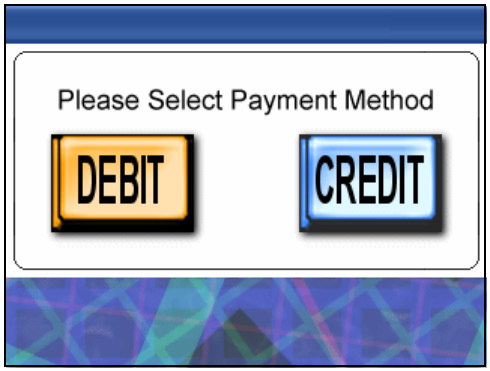
The **imgAd** control can be used to display advertisements. It includes five sample images that display a simple animated sequence. To display ads here simply replace the images with your own ads.

Use the **btnScrollUp** control to scroll through the contents of the **lstLineItems** control.

The **btnNext** control is used to go to the screen specified by the **scrLineItems.varScreenSkip** variable. The variable is blank by default. It can be set by the host ECR or by the script.

The **btnScrollDown** control is used to scroll down the contents of the **lstLineItems** control.

4.3 scrSelectPayment



The purpose of the scrSelectPayment screen is to enable the customer to select a payment type (debit or credit).

The scrSelectPayment screen contains five objects:

- Background: **bakBackground**
- Button: **btnDebit**
- Button: **btnCredit**
- Image List: **imgHeaderBar**
- Image List: **imgAd**

The **btnDebit** and **btnCredit** buttons direct the script to either the scrMsrDebit screen for debit transactions or the scrMsrCredit screen for credit transactions.

The **imgHeaderBar** control can be used to display a logo.

The **imgAd** control can be used to display advertisements. It includes five sample images that display a simple animated sequence. To display ads here simply replace the images with your own ads.

4.4 scrMsrCredit

The scrMsrCredit screen prompts the user to swipe their card.

The scrMsrCredit screen contains four objects:

- Background: **bakBackground**
- Image List: **imgHeaderBar**
- Image List: **imgSwipeCard**
- MSR: **msrCredit**

Use the **imgHeaderBar** control to display a logo.

The MSR control "**msrCredit"** is invisible. Because it's invisible, the size and placement of an MSR control is not important. It is only shown in the editor so it can be clicked to obtain its properties.

When the MSR control receives a successful card swipe, several events occurr.

1. PlaySystemSound(Success)
2. VariableSet(swipestatus,1)
3. FireEvent(swipestatus,1)
4. GotoScreenVariable(scrMsrCredit.varScreenOnSwipeGood)

When the MSR control receives a bad card swipe, several events occurr.

1. PlaySystemSound(Fail)
2. VariableSet(swipestatus,-1)
3. FireEvent(swipestatus,-1)
4. GotoScreenVariable(scrMsrCredit.varScreenOnSwipeBad)

When the MSR control exceeds the swipe limit these events take place:

1. PlaySystemSound(Fail)
2. VariableSet(swipestatus,-1)
3. FireEvent(swipestatus,-1)
4. GotoScreenVariable(scrMsrCredit.varScreenOnSwipeBad)

By default, the variables used in the **GotoScreenVariable** actions are either empty or set in the **scrWelcome** screen. An empty variable results in no screen jumps. The host ECR, the script, or an INI file can set the variables to cause a screen jump to take place.

4.5 scrMsrDebit

The scrMsrDebit screen prompts the user to swipe their card.

The scrMsrDebit screen contains four objects:

- Background: **bakBackground**
- Image List: **imgHeaderBar**
- Image List: **imgSwipeCard**
- MSR: **msrDebit**

Use the **imgHeaderBar** control to display a logo.

The "**msrDebit"** MSR control is invisible. Because it's invisible, the size and placement of an MSR control is not important. It is only shown in the editor so it can be clicked to obtain its properties.

When the MSR control receives a successful card swipe, several events occur

1. PlaySystemSound(Success)
2. VariableSet(swipestatus,1)
3. FireEvent(swipestatus,1)
4. GotoScreenVariable(scrMsrDebit.varScreenOnSwipeGood)

When the MSR control receives a bad card swipe, several events occur

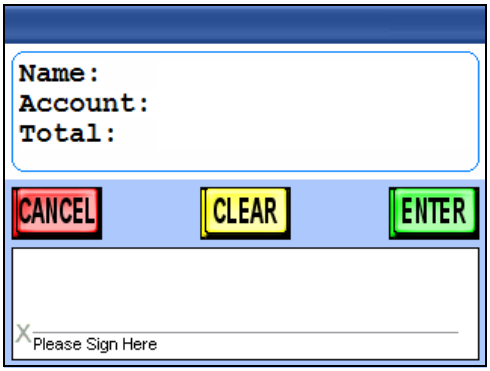
1. PlaySystemSound(Fail)
2. VariableSet(swipestatus,-1)
3. FireEvent(swipestatus,-1)
4. GotoScreenVariable(scrMsrDebit.varScreenOnSwipeBad)

When the MSR control exceeds the swipe limit, two events occur

1. PlaySystemSound(Fail)
2. GotoScreenVariable(scrMsrDebit.varScreenOnSwipeLimit)

By default, the variables used in the **GotoScreenVariable** actions are either empty or set in the **scrWelcome** screen. An empty variable results in no screen jumps. The host ECR, the script, or an INI file may set these variables to cause a screen jump to take place.

4.6 scrSignature



The scrSignature screen obtains a signature from the customer. scrSignature is normally used during credit transactions after the credit card was swiped.

The scrSignature screen **On entry** event performs the following actions:

1. FireEvent(ctstate,scrSignatureB)
2. SignatureBeginCapture(sigSignature)
3. VariableClear(btstate)
4. InputFieldAppendVariable(fldAccount,AccountNumber)

The **scrSignature** screen contains twelve objects:

- Background: bakBackground
- Signature: sigSignature
- Button: btnCANCEL
- Button: btnCLEAR
- Button: btnENTER
- Variable Text Label: txtFirstName
- Input Field: fldAccount
- Variable Text Label: txtTotal
- Static Text Label: lblAccount
- Static Text Label: lblTotal
- Static Text Label: lblName
- Image List: imgHeaderBar

The **sigSignature** control is set to use the SIG format for optimum signature quality.

The **btnCANCEL** button performs the following actions:

1. PlaySystemSound(Click)
2. SignatureClear(sigSignature)
3. VariableSet(btstate,btnCANCEL)
4. FireEvent(btstate,btnCANCEL)
5. GotoScreenVariable(scrSignature.varScreenCancel)

The **btnCLEAR** button performs the following actions:

1. PlaySystemSound(Click)
2. SignatureClear(sigSignature)
3. VariableSet(btstate,btnCLEAR)
4. FireEvent(btstate,btnCLEAR)

The **btnENTER** button performs the following actions:

1. PlaySystemSound(Click)
2. SignatureEndCapture(sigSignature)
3. VariableSet(btstate,btnENTER)
4. FireEvent(btstate,btnENTER)
5. GotoScreenVariable(scrSignature.varScreenEnter)

By default, the variables used in the **GotoScreenVariable** actions are either empty or set in the **scrWelcome** screen. An empty variable results in no screen jumps. The host ECR, the script, or an INI file may set these variables to cause a screen jump to take place.

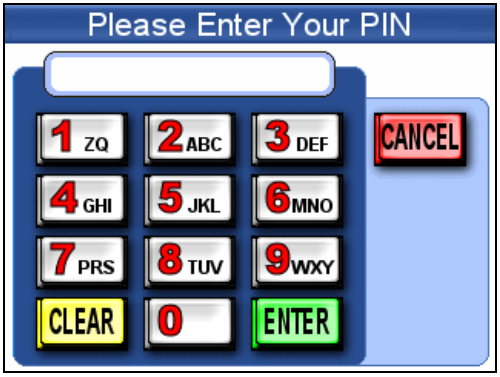
Use the **txtFirstName** Variable Text Label control to display the contents of the **FirstName** variable. The **FirstName** variable can be set by the host ECR or when a card is swiped containing the first name.

The **fldAccount** Input Field control is used to display the last four digits of an account number. The **fldAccount** Input Field control is set by an InputFieldAppendVariable action in the **On entry** event of the screen.

Use the **txtTotal** Variable Text Label control to display the contents of the **total** variable. The **txtTotal** variable can be set by the host ECR.

The editor does not simulate the signature actions.

4.7 scrPin



Use the purpose of the scrPin screen to obtain a Personal Identification Number or PIN from the customer. The scrPin screen is used during all debit transactions after a debit card is read.

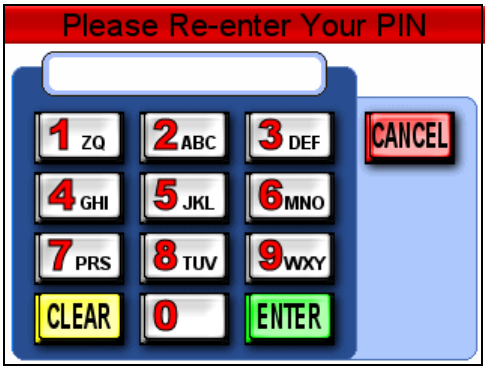
The scrPin screen contains 18 objects:

- PIN Pad: **pinPad**
- Pin Button: **btn1**
- Pin Button: **btn2**
- Pin Button: **btn3**
- Pin Button: **btn4**
- Pin Button: **btn5**
- Pin Button: **btn6**
- Pin Button: **btn7**
- Pin Button: **btn8**
- Pin Button: **btn9**
- Pin Button: **btn0**
- Pin Button: **btnCANCEL**
- Pin Button: **btnCLEAR**
- Pin Button: **btnENTER**
- Image List: **imgPleaseRe-enterYourPIN**
- Image List: **imgMessageArea**
- Variable Text Label: **txtMsgError**

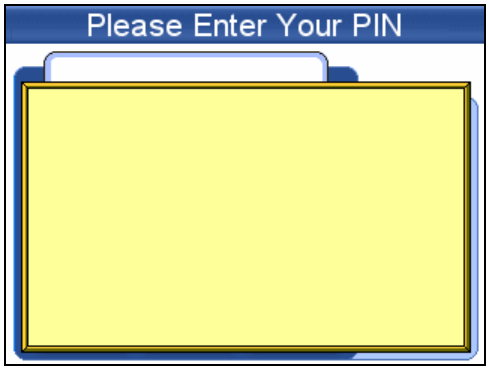
- Variable Text Label: **txtMsgProcessing**

The **pinPad** control utilizes several controls on the scrPin screen: **btn1**, **btn2**, **btn3**, **btn4**, **btn5**, **btn6**, **btn7**, **btn8**, **btn9**, **btn0**, **btnCANCEL**, **btnCLEAR**, and **btnENTER**. These buttons are special PIN button controls. The buttons are used for a PIN entry only. The buttons cannot be deleted and their positions and sizes are fixed. The button images and several properties of the **pinPad** (including the background image property) can be changed to enable a customized look and feel.

The **imgPleaseRe-enterYourPIN** control is normally invisible. It is displayed when an invalid PIN is entered and **btnENTER** is pressed. When displayed, it contains a red blinking message at the top of the screen prompting the user to re-enter their PIN.



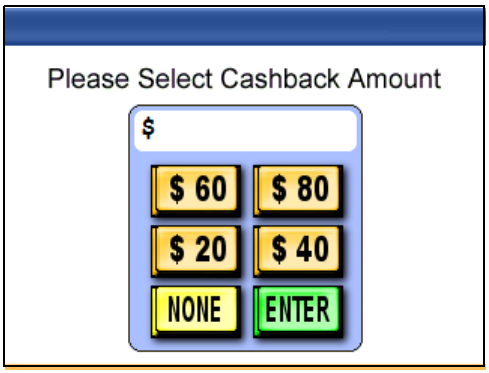
The **imgMessageArea** control is normally invisible. If a message is displayed, **imgMessageArea** is made visible so the text in either **txtMsgError** or **txtMsgProcessing** can be displayed over it:



The **txtMsgError** control is normally invisible. The control displays the contents of the **scrPin.varMsgError** variable. By default, the variable is set in the **scrWelcome** screen. The **pinPad** control **On error** event makes the **txtMsgError** and **imgMessageArea** controls visible.

The **txtMsgProcessing** control is normally invisible. The control displays the contents of the **scrPin.varMsgProcessing** variable. By default, the variable is set in the **scrWelcome** screen. The **pinPad** control **On processing** event makes the **txtMsgProcessing** and **imgMessageArea** controls visible.

4.8 scrCashBack



The purpose of the scrCashBack screen is to allow the customer to select a cash back amount. scrCashBack is normally offered to the customer during a debit transaction.

The scrCashBack screen contains ten objects:

- Background: **bakBackground**
- Image List: **imgHeaderBar**
- Button: **btn20**
- Button: **btn40**
- Button: **btn60**
- Button: **btn80**
- Button: **btnNONE**
- Variable Text Label: **txtCashBack**
- Static Text Label: **lblDollarSign**
- Button: **btnEnter**

Use the **txtCashBack** Variable Text Label control is used to display the current contents of the cash back system variable. The default value is $0.00 and can be changed by the ECR or by pressing one of the numeric cash back buttons. The numeric cash back buttons use the VariableSet action to set the cashback system variable.

4.9 scrThankYou

The purpose of the scrThankYou screen is to let the customer know the transaction has ended.

The **scrThankYou** screen is the only screen with a timeout by default. The default timeout is set to 5000 milliseconds. The timeout event contains one action:

GotoScreenVariable(scrThankYou.varScreenTimeout)

By default, the variables used in the **GotoScreenVariable** actions are either empty or set in the **scrWelcome** screen. An empty variable results in no screen jump action. The host ECR, script, or an INI file can set configure the variables to jump to another screen.

4.10 scrCustom

Use the scrCustom screen for creating custom screens. By default, it contains a Background Image.

When designing custom screens, Symbol recommends using standardized names for the screens.

Standard screen names include:

- scrWelcome
- scrLineItems
- scrSelectPayment
- scrMsrCredit
- scrMsrDebit
- scrSignature
- scrPin
- scrCashBack
- scrFinalTotal
- scrThankYou.

By using standard screen names, a host ECR application using the standard screen names works well with the script.

Note

The screen name can be changed in name field of **Screen Property** dialog box.

5 Toolbars, Screens, and Controls

posDesigner contains numerous **toolbars** enabling the developer to navigate the screens used to build point-of sale screen flows.

The posDesigner Screen and Control **panes** enable the developer to jump from one screen to another and select a control for editing.

Use posDesigner **controls** to define the look and behavior of the point-of-sale screen flow.

5.1 Control Property OK to All Option

If using more than one property window opened for a control, you are required to click “OK” once for each opened property window before the new settings take effect. In some cases, this amounts to several clicks. If you would like to bypass, use the “OK to All” option. Press “ALT” key on the keyboard while clicking the “OK” button on the current property window. This keyboard action causes all property windows to carry out their “OK” button actions automatically, thereby closing all the opened property windows.

5.2 Control Property Cancel All Option

If using more than one open property window for a control, click “CANCEL” once for each opened property window to cancel all modifications made to the property windows. In some cases, this amounts to several clicks. If you would like to bypass, use the “Cancel All” option. Press the “ALT” key on the keyboard while clicking the “CANCEL” button on the current property window. This keyboard action causes all property windows to carry out their “CANCEL” actions, thereby closing all opened property windows.

5.3 Controls Toolbar

The **Controls Toolbar** is divided into two sections. The section on the left contains the **Edit** tool group. The section on the right contains the **Creation** tool group.

Regardless of which tool is selected, clicking a screen control with the right mouse button always displays the edit menu options.

The Edit tool group consists of five tools:

Hand

Allows the developer to interact with controls and display their property sheets. Clicking a screen control with the left mouse button causes the editor POS terminal simulator to run the control On touch or On button action.

Selector

Allows the developer to select screen controls, move controls, resize controls, and display their property sheets.

1. To select a single control, click a control with the left mouse button.
2. To select multiple controls, press and hold the left mouse button to group the controls. When the desired controls are within the group, release the left mouse button.
3. To move one or more controls, select the control(s). Next, using the left mouse button, press and hold the center grab mark of the selected control(s). Move the control to the desired location and release the mouse button. Pressing the right mouse button while moving a control before releasing the left mouse button cancels the operation.
4. To resize one or more controls, select the control(s). Using the left mouse button, press and hold one of the 8 outer grab marks. Move the grab mark to the desired location to increase or decrease the size of the control(s). Release the mouse button. Press the right mouse button while resizing a control to cancel the operation

Eraser

To erase a control, press and release the left mouse button on the target control. To cancel the erase operation, move the cursor out of the screen and release the left mouse button.

Mover

To move a control, click the left mouse button on the target control and hold the mouse button until the object is moved to the desired location. Release the left mouse button again to confirm the new location. To cancel a move operation, press the right mouse button before releasing the left mouse button.

Resizer

To resize an object, move over the object with the mouse and press the left mouse button. Move the mouse to select the new size of the control. During the operation, the control anchor position is in the upper left-hand corner. While moving the mouse, the developer selects the new lower right-hand corner of the control. When done, press the left mouse button again to confirm the new size, or press the right mouse button to cancel the resize operation.

The **Creation** tool group consists of ten tools:

Button

Creates a new Button control.

Check Box

Creates a new Check Box control.

Radio Button	Creates a new Radio Button control.
Static Text Label	Creates a new Static Text Label control.
Variable Text Label	Creates a new Variable Text Label control.
Input Field	Creates a new Input Field control.
Image List	Creates a new Image List control.
Signature	Creates a new Signature control.
Line Display	Creates a new Line Display.
MSR	Creates a new MSR control.

5.3.1 Creating a New MSR Control

Only one MSR can be added to a screen. Once one MSR control is present in a screen, the editor does not allow a second MSR control to be added.

To create a new control:

1. Select the control to be created.
2. Move the mouse to the position where the upper left-hand corner of the control is to be placed and press the left mouse button.
3. Move the mouse to where the lower right-hand corner of the control is to be placed.
4. Press the left mouse button again to confirm the operation, or press the right mouse button to cancel the operation.

5.4 Grid Toolbar

The posDesigner editor was developed to simplify screen design. To aid in screen design, the grid guide can be used to help align controls on a screen. When enabled, the grid guide displays a series of horizontal and vertical guidelines. The spacing, margins, color, and appearance of these grid lines can be changes by setting the Grid guide properties. Additionally, an optional **Snap to Grid** mode can be enabled. When Snap to Grid is enabled, the mouse cursor can create, move, and resize controls as if the mouse cursor was exactly aligned within the grid guidelines, even when the mouse cursor is not. This makes it simple to perfectly align your controls.

The Grid tool group consists of three tools:

Grid Guide	Enables or disables the Grid guide.
Grid Option	Edits the Grid guide properties.
Snap to Grid	Enables/disables the Grid guide Snap to feature.

5.5 Alignment Toolbar

Use the Alignment Toolbar when designing screens by performing variable alignment operations on the current selected control.

Align Horizontal	Aligns the current control to the horizontal center of the screen.
Align Vertical	Aligns the current control to the vertical center of the screen.
Align Center	Aligns the current control to the center of the screen.
Align Left	Aligns the current control to the left of the screen**.**

Align Right	Aligns the current control to the right of the screen.
Align Top	Aligns the current control to the top of the screen.
Align Bottom	Aligns the current control to the bottom of the screen.

For more information on using Snap to Grid when aligning a control, refer to Grid Toolbar on page *Grid Toolbar* on page 5-5.

5.6 Main Toolbar

The **Main** toolbar is divided into five sections: **File**, **Clipboard**, **Undo/Redo**, **Print** and **Help**.

New	Creates a new document.
Open	Opens an existing document.
Insert	Inserts an existing document to the current one.
Import	Imports an IFB file.
Save	Saves the current document.
Save Screen	Saves the current screen.
Export	Exports an IFB file.
Upload	Uploads an IFB file.
Cut	Cuts the current screen.
Copy	Copies the current screen.
Paste	Pastes a cut/copied screen into the screen.
Undo	Undoes the last screen editing operation.
Redo	Redoes the last undone screen editing operation.

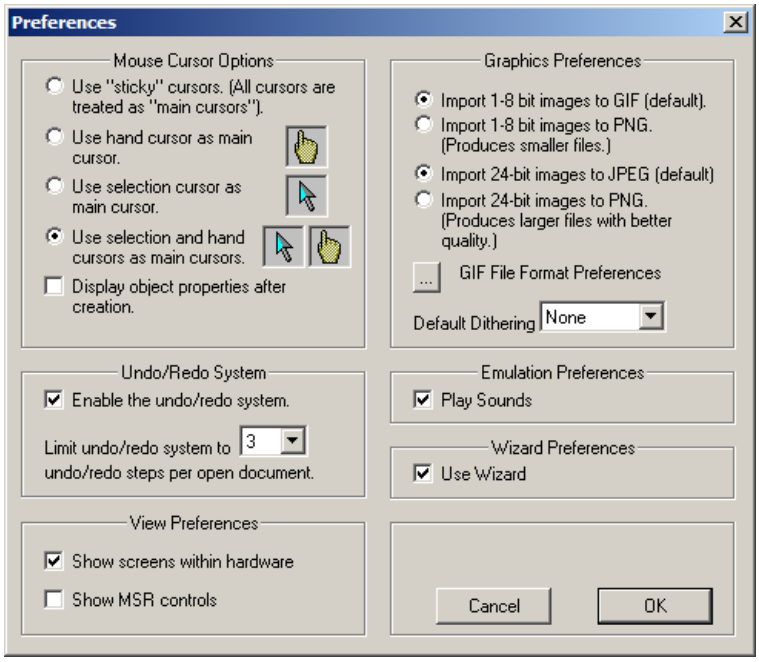
Print Prints the current screen.

Help About Displays the About dialog box.

Note

If saving a screen to the default location using a default screen name (such as scrWelcome, scrLineItems etc.), the screen replaces the previous default screen as a new default screen.

File + Preferences enables the developer to change the settings for the Undo/Redo main toolbar features.


Preferences
Mouse Cursor Options
Use "sticky" cursors. (All cursors are treated as "main cursors").
Use hand cursor as main cursor.
Use selection cursor as main cursor.
Use selection and hand cursors as main cursors.
Display object properties after creation.
Graphics Preferences
Import 1-8 bit images to GIF (default).
Import 1-8 bit images to PNG. (Produces smaller files.)
Import 24-bit images to JPEG (default)
Import 24-bit images to PNG. (Produces larger files with better quality.)
GIF File Format Preferences
Default Dithering
None
Undo/Redo System
Enable the undo/redo system.
Limit undo/redo system to
3
undo/redo steps per open document.
Emulation Preferences
Play Sounds
Wizard Preferences
Use Wizard
View Preferences
Show screens within hardware
Show MSR controls
Cancel
OK

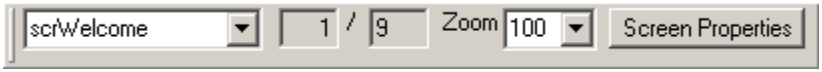
5.7 Navigation Toolbar

Use the **Navigation** toolbar to navigate the screen emulator and enable or disable screen animations and screen actions.

The Navigation toolbar contains 6 controls:

Start	Moves the screen emulator to the first screen.
Previous	Moves the screen emulator to the previous screen.
Next	Moves the screen emulator to the next screen.
End	Moves the screen emulator to the last screen.
Play	Enables animations and button actions to be emulated
Stop	Disables animations and button actions to be emulated.

5.8 Screen Toolbar



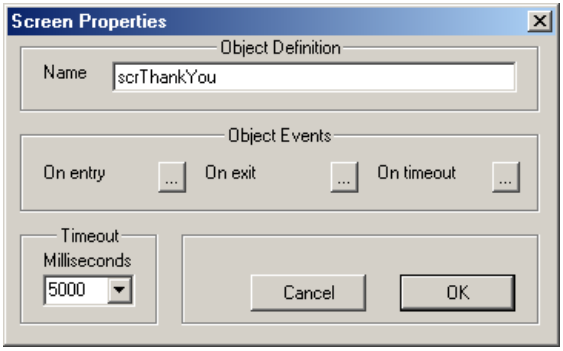
The **Screen** toolbar contains the Screen Field, the Screen Location indicators, the Zoom Field and the Screen Properties tools.

- **Screen Field —** Used to navigate to a desire screen and rename a screen. To navigate to a desired screen, select the drop down button and click the desired screen. To rename the current screen, type in the new name and press enter.
- **Screen Location** — Used to show which screen is currently being viewed in the emulator. The number on the left is the current screen index. The number on the right is the total number of screens present in the current document.

- **Zoom Field** — Used to control the zoom feature. To zoom in, select a zoom percentage above 100%. To zoom out, select a zoom percentage below 100%. A zoom level of 100% displays a pixel per pixel accurate representation of the POS terminals display being emulated.
- **Screen Properties** — Used to control the properties of a screen such as: the screen name, the On entry action, the On exit action, the On timeout action and the timeout value.

5.9 Screen Properties

Click the **Screen Properties** button to display the Screen Property dialog box. The Screen Property dialog box can also be opened by double-clicking the screen name in the Screen Pane.

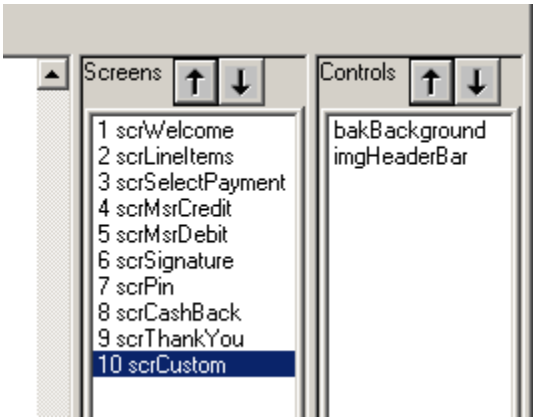


Editable screen properties include: screen name, On entry event, On exit event, On timeout event, and timeout value.

The **On entry** event is triggered when the screen is loaded. The **On exit** event is triggered when the screen exits.

The Screen Timeout field is set to **ZERO** for most screens. If set to ZERO, the screen timeout function is disabled for the current screen and the Timeout Actions for the **On Timeout** event are ignored. Setting the Screen Timeout value to a non-zero value enables the screen Timeout Actions for the current screen. If enabled, the Screen Timeout field indicates how many milliseconds should elapse before the screen timeout is reached. When the screen timeout is reached, the POS application performs actions for setting the On Timeout event of the current screen.

5.10 Screens Pane



The **Screens Pane** enables the developer to jump from one screen to another and change the screen order. The Screens Pane is located near the right-hand side of the editor on the left of the Controls Pane.

A list of the present screens is displayed in the Screen Pane.

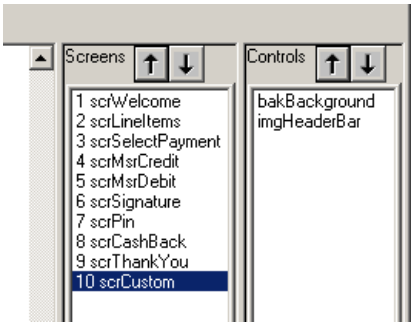
To select and edit a screen, use the left mouse button to click the screen name listed in the Screen Pane.

To move a screen up or down, use the UP or DOWN arrow buttons located at the top of the Screen Pane. Moving a screen up or down does not impact the screen flow. Screen flow is controlled by actions present in controls on the screen. Moving a screen up or down changes the order in which they are stored.

Double clicking a screen in the Screen Pane displays the screen's property dialog box.

5.11 Controls Pane

Use the Controls Pane to select a control for editing. The Control Pane enables the developer to change control z-order placement in the current screen. The Controls Pane is located on the right-hand side of the editor.



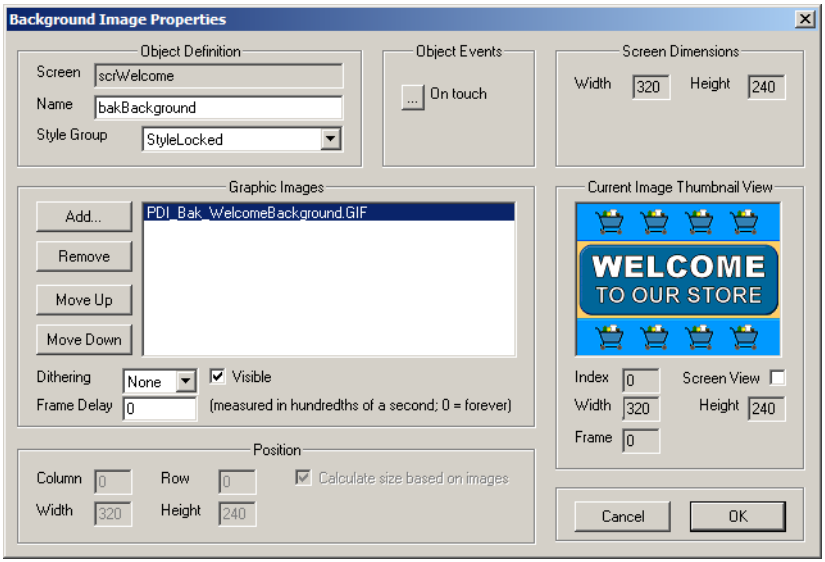
The Controls Pane displays a list of all the controls present in the current screen.

To edit the properties of a control, double-click the control name listed in the Controls Pane using the left mouse button.

To select a control, single-click the control name listed in the Controls Pane using the left mouse button.

To change the z-order of any selected control, use the UP or DOWN arrow buttons located at the top of the Controls Pane. The first control listed at the top of the list is the control with a z-order of ZERO, meaning all other controls are drawn over it. The control at the bottom of the list is drawn last and has the highest z-order position, this control overlays all other controls.

5.12 Background Image Control



The **Background Image** control is present within all default screens. It provides a screen with a background image. The **Background Image** control should be the first control present in a screen.

To change the Background Image of a screen, open the **Background Image Properties** dialog box by either right-clicking on the Background Image control in the emulator view or by double-clicking the Background Image control name in the Control Pane.

Editable properties include: object **Name**, object **Style Group**, **On touch** event, **Visible** property, **Frame Delay**, **Dithering**, and the graphic images present in the object.

The **Style Group** allows you to select a style group for your image. The default group is "StyleLocked". For more information, refer to *Chapter 6, posDesigner Styles*.

The **On touch** event enables the creation of an action occurring when the background image is touched.

By default, a background image control contains only one graphic image. But it can also contain more than one graphic image. To add graphic images, press the **Add…** button. To remove graphic image(s), select the image(s) to remove and press **Remove**. To change the order images are displayed, select the image(s) to re-order and press either **Move Up** or **Move Down** to change the order they appear.

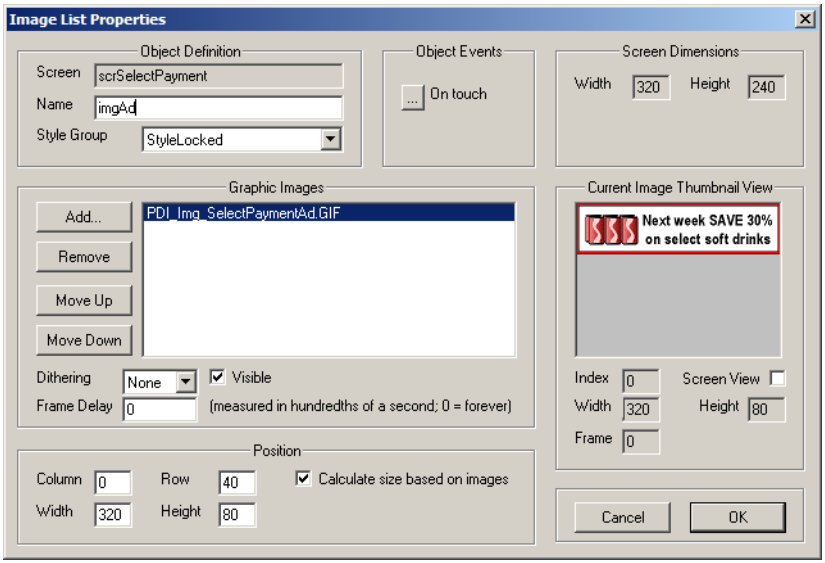
By default, the **Frame Delay** of a background image is **ZERO**. Zero disables animation. When there is more than one graphic image present in a background image control, enable animation by setting the **Frame Delay** rate to any non-zero value. This causes all images in the list to be displayed one after the next at the rate specified. After the last image is displayed, the next image displayed is the first image in the list.

For more information, refer to *Colors, Resolution, and Dithering* on page 2-2.

The **Visible** checkbox is checked by default. This allows the control to be seen. If unchecked, the control is invisible, but still active. The developer can touch a background image control, visible or not. When the control is invisible, it is not shown in the emulator but appears in the **Controls Pane**. Even if the control is displayed in the editor, you can still select and edit the control.

5.13 Image List Control

The **Image List Control** is contained in numerous screens. It provides a screen with an area to display advertisements and other graphic images overlaying the background.



To change an Image List control, open the **Image List Properties** dialog box by either right-clicking on the Image List control in the emulator view or double-clicking the Image List control name in the Control Pane.

Editable properties include: object **Name**, **Style Group**, **On touch** event, **Visible** property, **Frame Delay**, **Dithering**, and graphic images present in the object.

The **Style Group** allows you to select a style group for your image. The default group is "StyleLocked". For more information, refer to *Chapter 6, posDesigner Styles*.

The **On touch** event enables the creation of an action occurring when the image is touched.

By default, an image list control contains only one graphic image. But it can also contain more than one graphic image. To add graphic images, press **Add…**. To remove graphic image(s), select the image(s) and press **Remove**. To change the order images are displayed, select the image(s) to re-order and press either **Move Up** or **Move Down** to change the order they appear.

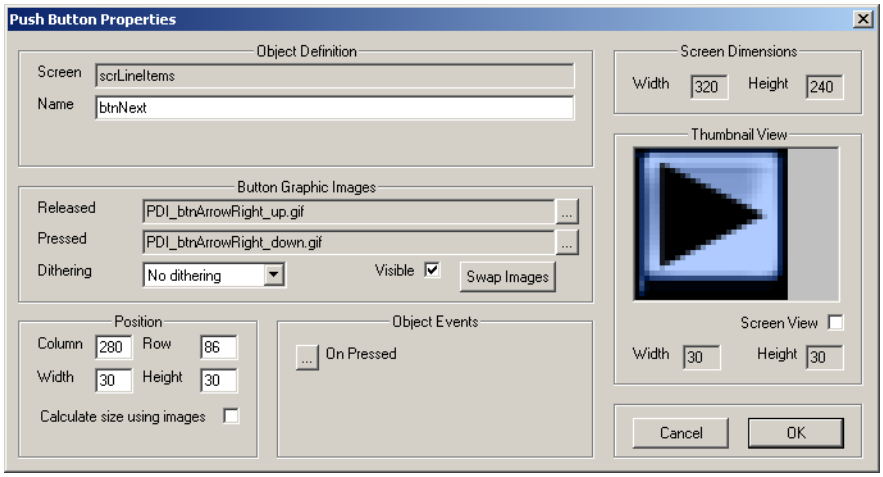
By default, the **Frame Delay** of an image list control is set to **ZERO**. This disables animation. When there is more than one graphic image present in an image list control, enable animation by setting the **Frame Delay** rate to any non-zero value. This causes all images in the list to be displayed one after the next at the rate specified. After the last image is displayed, the next image displayed is the first image in the list.

For more information, refer to *Colors, Resolution, and Dithering* on page 2-2.

The **Visible** checkbox is checked by default. This allows the control to be seen. If unchecked, the control is invisible, but still active. The developer can touch an image list control whether visible or not. When the control is invisible, it is not shown in the emulator but appears in the **Controls Pane**. Even though the control is now shown in the editor, you can still select and edit the control.

5.14 Button Control

The **Button Control** enables the developer to interact with the screen flow of a POS terminal, giving feed back to questions presented, causing a screen to jump from one screen to another, and more.



To change a Button control on a screen, open the **Push Button Properties** dialog by either right-clicking on the Button control in the emulator view or double-clicking the Button control name in the Control Pane.

Editable button properties include: object name, size and position of the object, graphic images present in the object and the On button pressed event.

On button pressed events create an action that occurs when the Button control is pressed. All buttons have two images associated with them. The **Image Up** image displays when the Button control is not pressed. The **Image Down** image displays when the Button control is being pressed.

For more information, refer to *Colors, Resolution, and Dithering* on page 2-2.

The **Visible** checkbox is checked by default. If unchecked, the button is invisible, but active. The developer can press a button whether visible or not. When the control is invisible, its location is displayed in the emulator using a dotted outline. This outline is not seen when the screen is displayed on the POS terminal.

The **Position** section includes button size and position information. The **Calculate size based on images** option is unchecked by default, forcing the image to stretch or shrink to meet the size of the button size. If this option is checked, the button size is adjusted to match to the size of the image.

5.15 Check Box Control

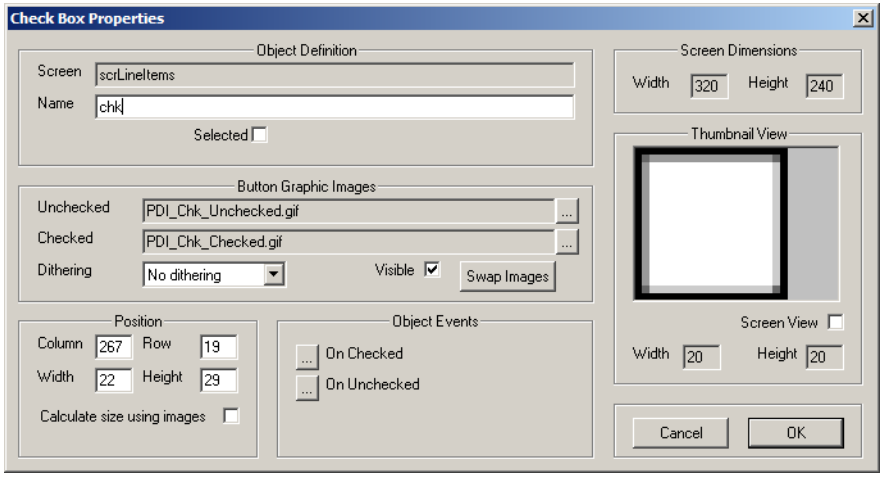
The **Check Box Control** contains one or more items that display checked when selected. The Check Box control enables users to provide feed back to questions. For a group of Check Box Controls, select multiple items at one time.

To change a Check Box control, open the **Check Box Properties** dialog box by either right-clicking on the Check Box control in the emulator view or by double-clicking the Check Box control name in the Control Pane.

Editable properties of a Check Box include: object name, default object status, object size and position, graphic images present in the object, and the On button selected event.

The **On Checked** event allows the creation of an action occurring when the Check Box control is checked.

The **On Unchecked** event allows the creation of an action occurring when the Check Box control is unchecked.



All check boxes have two images associated with them. The **Image Up** image is displayed when the Check Box control is not selected. The **Image Down** image displays when the Check Box control is being selected.

The **Selected** checkbox is not checked by default. This means the Image Up image displays by default. If the Selected field is checked, the Image Down image displays by default.

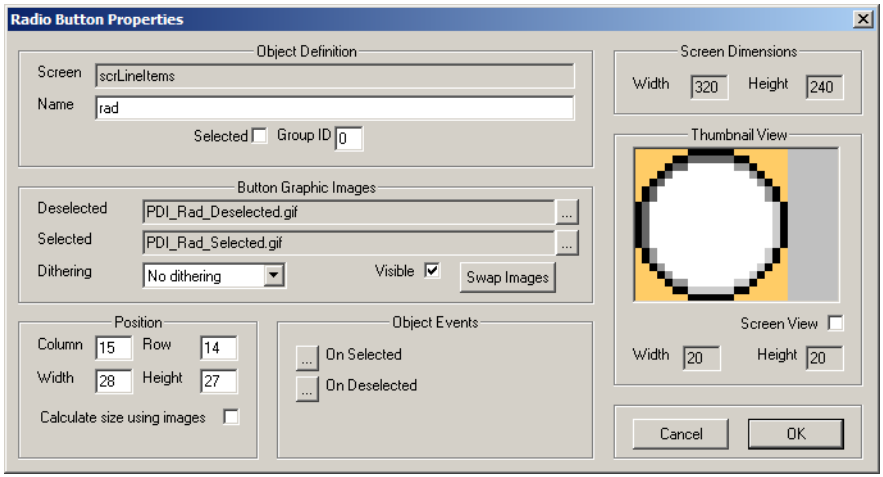
The **Visible** field is checked by default. If unchecked, the checkbox is invisible, but still active. When the control is invisible, its location is displayed in the emulator using a dotted outline. The outline is not seen when the screen displays on the POS terminal.

For more information, refer to *Colors, Resolution, and Dithering* on page 2-2.

The **Position** field includes the size and position information of the checkbox. The **Calculate size based on images** option is unchecked by default, forcing the image to stretch or shrink to fit the size of the checkbox. If checked, the size of the checkbox is adjusted to fit to the size of the image.

5.16 Radio Button Control

The Radio Button Control contains one or more items displaying as a group. The Radio Button Control enables users to give feedback to questions. Users can only select one item at one time, making radio buttons mutually exclusive.



To change a Radio Button control, open the **Radio Button Properties** dialog box by either right-clicking on the Radio Button control in the emulator view or by double-clicking the Radio Button control name in the Control Pane.

Editable properties of a Radio Button include: object name, default object status, object size and position, the graphic images present in the object and the On button selected event.

The **On Selected** event enables the creation of an action occurring when the Radio Button control is selected.

The **On Deselected** event enables the creation of an action occurring when the Radio Button control is deselected.

Radio buttons have two images. The Image Up image displays when the Radio Button control is not selected. The Image Down image displays when the Radio Button control is selected.

The **Selected** checkbox unchecked by default. The Image Up image displays by default. If the Selected checkbox is selected, the Image Down image displays.

The **Group** checkbox indicates which group the current radio button belongs to. In one group, there could be several mutually exclusive radio buttons. Only one radio button can be selected at one time.

The **Visible** checkbox is checked by default enabling the radio button to be visible. If unchecked, the radio button is invisible, but still active. Optionally select a radio button whether it is visible or not. When the control is invisible, its location displays in the emulator using a dotted outline. The outline is not seen when the screen is displays on the POS terminal.

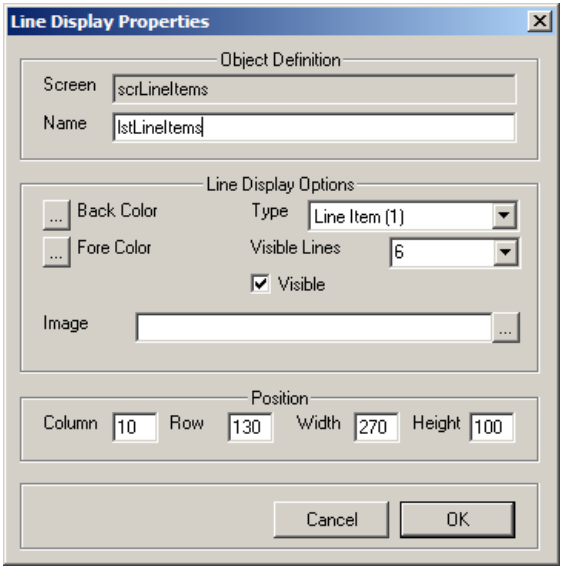
For more information, refer to *Colors, Resolution, and Dithering* on page 2-2.

The **Position** field includes radio button size and position information. The **Calculate size based on images** option is unchecked by default, forcing the image to stretch or shrink to fit the size of the radio button. If the option is checked, the size of the radio button is adjusted to fit to the size of the image.

5.17 Line Display Control

Use the **Line Display Control** to display a list of scrolling line items on the POS terminal. The Line Display Control can also be used to display a list of custom text data.

To change a Line Display control, open the **Line Display Properties** dialog box by right-clicking on the Line Display control in the emulator view or double-clicking the Line Display control name in the Control Pane.

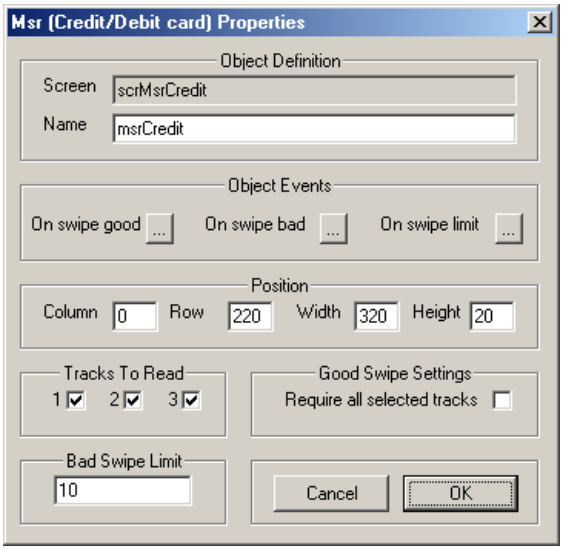


Editable properties of a Line Display include: object name, object size and position, the graphic image present in the object, object visibility, colors used, number of lines to be displayed, and the type of data the Line Display control should display.

The default Type for the Line Display control is **Line Item (1)**. This causes the control to display type 1 line items passed from the ECR when running on a POS terminal. User settings such as User Data (2) are for non-line item use, but can be used for any data (including line items). The LineDisplayAdd, LineDisplayClear, LineDisplayScrollDownb, and LineDisplayScrollUp actions can be used with this control.

Refer to *Chapter 8, posDesigner Actions* for additional information.

5.18 MSR Control



The **MSR** (Magnetic Stripe Reader) Control is used to capture credit card and debit card data. Any screen containing a MSR control is capable of capturing card data.

To change an MSR control, open the **MSR Properties** dialog box by right-clicking on the an MSR control in the emulator view or double-clicking the MSR control name in the Control Pane.

Editable properties of the MSR control include: object name, size and position of the object, number of tracks to read, whether or not to require all selected tracks for a good read, bad swipe limit and the On swipe good, On swipe bad and On swipe limit events.

The MSR control is an invisible control. In the emulator, the MSR control is displays using a dotted outline. This is not visible on the POS terminal. It only displays in the emulator to allow its properties to be edited by right-clicking the control. The size and position of the MSR control is ignored when the control is on the POS terminal.

Use the **Tracks To Read** checkboxes to select the tracks the MSR control reads from a card. Any combination of tracks 1, 2, and 3 can be selected. Select the **Require all selected tracks** checkbox to generate an On swipe good event when all selected Tracks To Read tracks are successfully read. If the Require all selected tracks option is not selected, any selected track successfully read generates an **On swipe good** event, even if the other selected tracks are not successfully read.

On swipe good events take place when a card is swiped and its data successfully read from the card magnetic stripe.

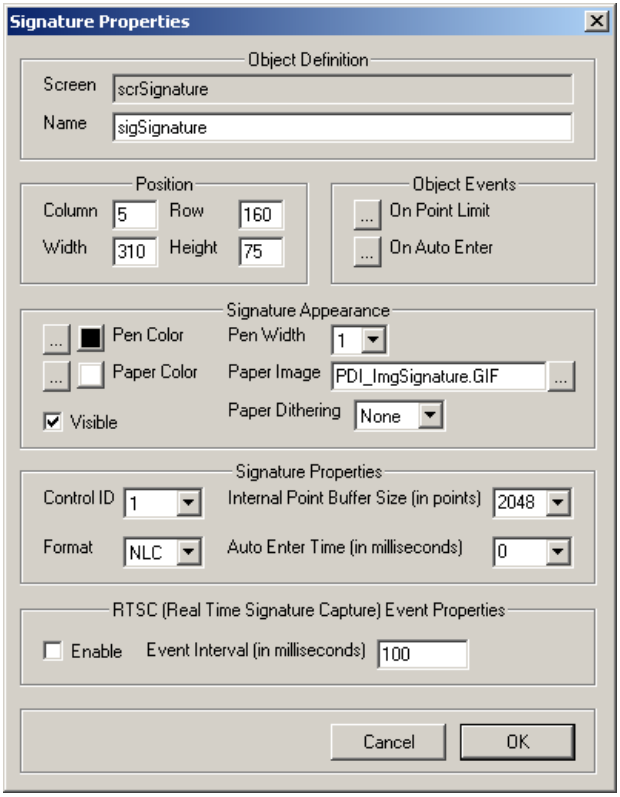
On swipe bad events take place when a card is swiped, but the data was not successfully read from the card magnetic stripe.

On swipe limit events take place when the **Bad Swipe Limit** is reached. If the Bad Swipe Limit is set to zero, the action never takes place.

If the **Good Swipe Settings** checkbox is selected, all the Tracks To Read checked should be satisfied.

The **Bad Swipe Limit** value has a default of 10. The MSR control allows up to 10 consecutive bad swipes before triggering **On swipe limit** events. Setting the value to zero disables the On swipe limit events.

5.19 Signature Control



Signatures are used by retailers to verify credit transactions. The storage and retrieval of paper signatures is an expensive process for retailers. Having a paperless digital signature on record saves both money and time when compared with signatures stored on paper.

To change a Signature control, open the **Signature Properties** dialog box by right-clicking on the Signature control in the emulator view or double-clicking the Signature control name in the Control Pane.

Editable properties of a Signature control include: object name, size and position of the object, pen and paper color, paper image, visibility, format, pen width, auto enter time, signature point limit and the On Point Limit and On Auto Enter events.

Paper Color allows you to select the paper color of the signature control. In a **Paper Image**, this setting is not used.

Pen Color allows you to select the color of the pen used to draw a signature in the signature control.

Paper Image allows you to supply a graphic image to represent the paper used in the signature control. Pen strokes always display on top of the paper.

On Point Limit events take place when the pen movements captured in the Signature control reach the Point Limit value specified.

On Auto Enter events occur when the Auto Enter timeout is reached. The **Format** option allows the selection of the signature file format used to store a captured signature.

The **Auto Enter** option specifies the timeout value used to automatically end signature capture. When set to zero, the auto enter feature is disabled. If set to 15, a total of 15 seconds of stylus absence elapses after signing in the screen signature area before the **On Auto Enter** actions take place. Writing in the signature area with the stylus sets the timeout counter to zero. The timeout counter increases once every second after the stylus is lifted from signature area. If the signature area is never touched with the stylus, the timeout counter stays at zero and the On Auto Enter events never take place.

The **Point Limit** option specifies the maximum number of stylus movements to capture in the Signature control.

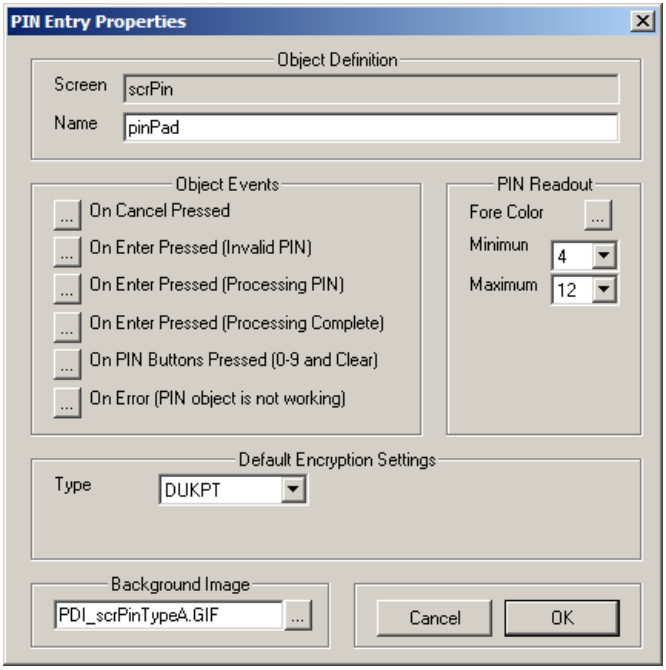
5.20 PIN Control

The **PIN** control is used to enter a PIN (Personal Identification Number) during debit transactions. The PIN control occupies the entire screen. It contains fixed buttons and a text field that cannot be moved or deleted.

To change a PIN control present on a screen, open the **PIN Entry Properties** dialog box by right-clicking on the PIN control in the emulator view or double-clicking the PIN control name in the Control Pane.

Editable properties of a PIN control include: object name, encryption type, background image, button images, valid length of PIN, color of readout text, and six events: On Processing, On Completion, On Button Press, On Invalid Entry, On Cancel, and On Error.

The buttons on a PIN control are restricted Button controls. The developer can only change the button images of a PIN control's buttons. For more information, refer to *Button Control* on page 5-14.



The **Encryption Type** option enables the developer to select the type of encryption to encrypt the PIN. Both **DUKPT** and **MASTER/SESSION** encryption types are available.

When the **Type** option is **Master/Session** and your script is deployed, the Master Key Index and Session Key information are required to be sent from the cash register at run-time. On the PD8500, the Master Key should always be set to 0. The Session Key should always be a 16 digit hexadecimal number.

On Cancel Pressed events occur when **CANCEL** is pressed. The default action is for the screen flow to move to the **scrSelectPayment** screen. **On Error** events take place when the PIN encryption fails. The default action is for the screen flow to move to the **scrSelectPayment** screen.

On Enter Pressed (Invalid PIN) events take place whenever any invalid PIN is entered and the **ENTER** button is pressed. The default action plays an INVALID sound (a base beep).

On Enter Pressed (Processing PIN) events take place when the **ENTER** button on a PIN control is pressed. When a valid PIN is entered, the default is no action. Symbol recommends creating a hidden Image List control containing a **Processing, Please Wait …** message visible.

On Enter Pressed (Processing Complete) events take place when the PIN has successfully been encrypted and the new encrypted PIN is available for the ECR. The default action is for the screen flow to move to the **scrCashBack** screen.

On PIN Buttons Pressed (0-9 and Clear) events take place whenever any 0-9 or Clear PIN button is pressed. The default action plays a button click sound. All PIN buttons generate On Button Press events.

On Error (PIN object is not working) events take place whenever the PIN object is not functioning properly. The default action is "GotoScreen scrSelectPayment".

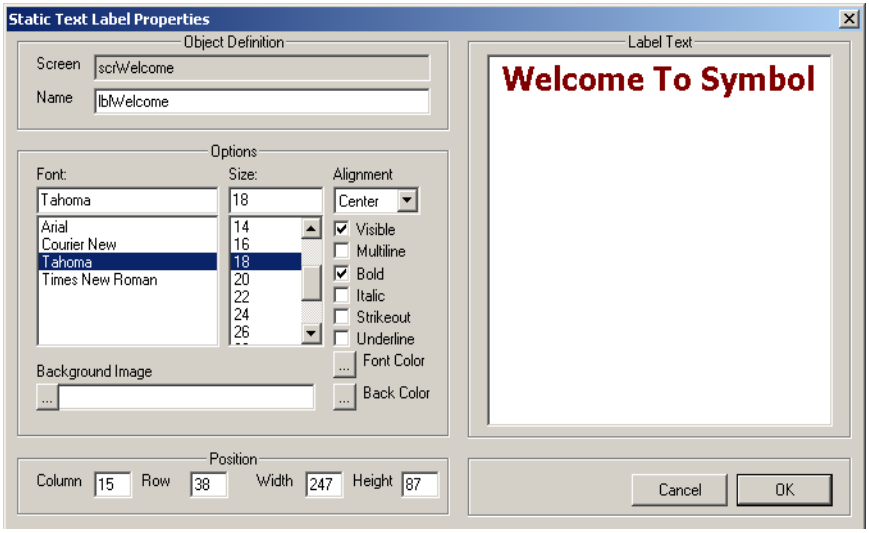
The **PIN Digit Length** maximum and minimum values can be modified. If (for example) a minimum of 4 digits and maximum of 6 digits is selected, a PIN of 3 digits or less or 7 digits or more is not a valid PIN.

Note

In order to encrypt a pin number successfully, a valid card account number should be captured in advance.

If setting encryption type is MASTER/SESSION (M/S) from the host side, the Master Key Index (mindex) and Session Key (seskey) should also be provided from the cash register. Otherwise, DUKPT is used.

5.21 Static Text Label Control

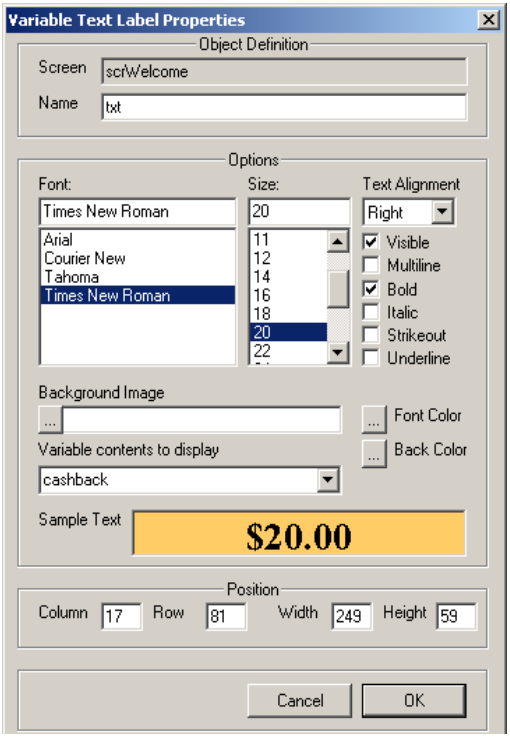


The **Static Text Label Control** is used to display text data. To change a Static Text Label control present on a screen, open the Static Text Label Properties dialog box by either right-clicking the Static Text Label control in the emulator view or double-clicking the Static Text Label control name in the Control Pane.

Editable properties of the Static Text Label control include: object name, background image, color, size and position of the control, text alignment, left-input option, multi-line option, visibility, and Label Text.

Label Text is used to set the text displayed in the label.

5.22 Variable Text Label Control



Use the **Variable Text Label Control** to display the current text data contents the associated variable it is bound with. When the contents of the associated variable changes, the Variable Text Label is updated to display the new contents.

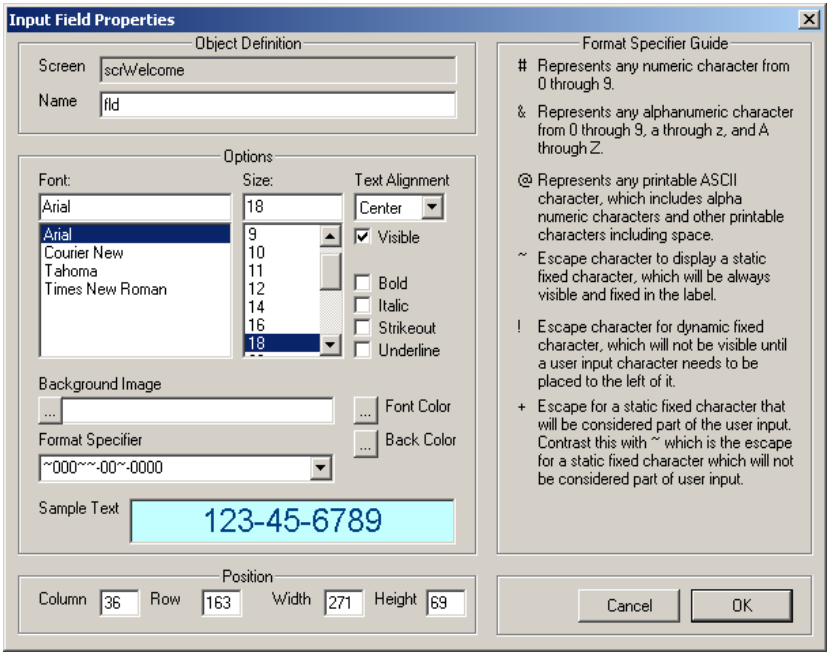
To change a Variable Text Label control, open the Variable Text Label Properties dialog box by either right-clicking on the Variable Text Label control in the emulator view or double-clicking the Variable Text Label control name in the Control Pane.

Editable properties of Text Variable Label control include: object name, background image, color, size, and position of the control, text alignment, left-input option, multi-line option, visibility, and variable contents to display.

Variable contents to display is the name of the variable the developer displays in the label.

For a list of standard POS variables, refer to the posPortal CE Integrator Guide (72E-64639-02).

5.23 Input Field Control



Use the **Input Field Control** to create input fields such as phone number entry fields and date entry fields.

To change an Input Field control, open the **Input Field Properties** dialog box by either right-clicking on the Input Field control in the emulator view or double-clicking the Input Field control name in the Control Pane.

Editable properties of Input Field control include: object name, background image, color, control size and position, text alignment, left-input option, multi-line option, visibility and Format Specifier.

The Input Field accepts text input from the InputFieldAppend and InputFieldAppendVariable actions and can be cleared using the InputFieldClear action. Its contents can be saved to a variable using the InputFieldSave action. These actions can be part of a button sending the text strings to the input field when pressed.

Use the Format Specifier to specify how to display the data in the Input Field. For example, to display a 10 digit phone number, use "*#~-###~-###~-####*". By adding the string "1234567890" (using InputFieldAppend) the Input Field displays "1-234-567-890".

5.24 Installation File Block (IFB)

IFB stands for Installation File Block.

- **File + Export IFB** enables the document to be saved as an .ifb file for later use.
- **File + Import IFB** enables the import of an existing .ifb file as the current document.
- **File + Upload IFB** enables the user to load an IFB file on the posPortal device.

5.25 Running a Script on the POS Terminal

When the screen flow design is complete, Symbol recommends testing the flow within posDesigner before running the script on the Symbol POS interactive terminal.

To run a script on a POS interactive terminal:

1. Connect posDesigner to the Symbol POS interactive terminal using a USB cable, RS232 cable, or an Ethernet cable. Wait for the device to boot up completely.
2. From posDesigner, select **File+Upload IFB**.

   The IFB property dialog box displays. Check "Erase previous folder contents" to erase any previous script data on the terminal.
3. Press the OK button.

   If Creating a non-VISA PED script, proceed to step 4.

   a. A prompt displays "Digitally sign IFB?". When creating a secure VISA PED, digitally signing the IFB is a requirement.
   b. Press "Yes".
   c. A message displays promting for the "Key Manager #" to enter a password (there is a total of 2 or 3 key managers).
   d. Insert the floppy disk containing the digital key into the floppy drive.
   e. Type the “Key Manager #" unique password and press OK.
   f. If a message displays prompting for the next "Key Manager" to enter a password, go to substep d and repeat the process until no more "Key Mangers" are required to sign the IFB.

   The "Upload script" dialog box displays.
4. Select the correct device connection type. Press the "Upload" button.

   Upload status information displays in the text area and posDesigner sets the unit to download mode and begins transferring the new IFB to the Symbol POS interactive terminal.
5. When the upload is complete, press "Done" to exit the "Upload script" dialog box.

   The Symbol POS interactive terminal automatically reboots and runs the new script.

If the posBuilder Win32 SDK (version 6) is installed, test the script using Tools>Host Test. This allows you to run screens, set variables, and get variables present on the Symbol POS interactive terminal using posDesigner. Script testing uses the Symbol posBuilder Win32 SDK. Refer to the posBuilder W32 Software Development Toolkit (available at *http://software.symbol.com/devzone*) for more information on how to use posBuilder.

6

posDesigner Styles

The posDesigner software allows the developer to create styles to apply to preexisting scripts. A “Style” is similar in concept to a “Skin”. A “Style” is package containing a collection of images used to replace existing images in a script.

The PD8500 VisaPED POS terminal ships complete with a default VisaPED complient script. This script normally supports posDesigner “Style” packages. Some developers choose not to create their own scripts, but instead use the built in default script and create “Style” package to apply to it.

While the VisaPED script itself is secure and must be digitally signed before it can be used in the PD8500 VisaPED POS terminal, a “Style” package does not need to be digitally signed. The PD8500 VisaPED POS and non-VisaPED POS terminals accepts any uploaded “Style” package. It is up to the script to decide which images it replaces with the images present in a “Style” package. Each stylizable control in a script has its own style group property. If all the controls in a script use the style group “StyleLocked” then a style cannot be applied to the script. For security reasons, the default VisaPED scripts only allow certain controls to be stylized.

When a script is created, the developer can choose which images are stylizable and which are not. When creating a new control, the default style group of the control is "StyleLocked". This prevents the control from being stylized by a Style package. To allows a control to be stylized, the devoloper much change the style group fo the control to any setting other than "StyleLocked".

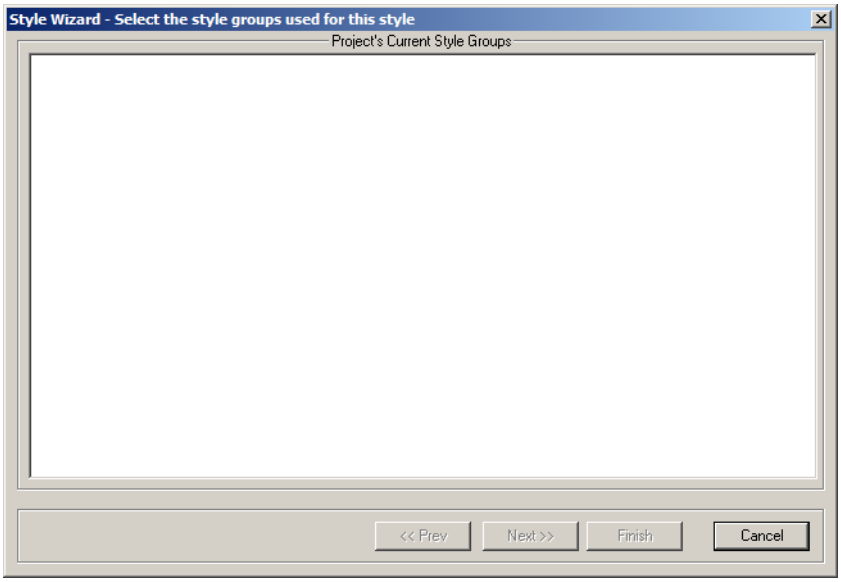
The following style groups are available for stylizable controls:

StyleLocked	This is the default style group. This group prevents a control from being stylized. In the default VisaPED script, all secure controls (such as the cash back screens and most buttons) use this style group to prevent styles from being applied to them.
StyleCorporate	This group is used for logos. In the default VisaPED script, all stylizable header bars use this style group.
StyleAdvertisement	This group is used for advertisements. In the default VisaPED script, all stylizable image list controls used for ad display use this style group.
StyleSeasonal	This group is used for seasonal images. For example, some images in a script may change in summer, then again in fall. Such images should belong to this group. The default VisaPED script uses this style group for the background images of many screens such as the scrWelcome and scrThankYou screens.
StyleDaily	This group is used for images that change daily. The default VisaPED script does not use this style group.
StyleWeekly	This group is used for images that change daily. The default VisaPED script does not use this style group.
StyleMonthly	This group is used for images that change monthly. The default VisaPED script does not use this style group.
StyleAnnual	This group is used for images that change yearly. The default VisaPED script does not use this style group.
StylePerennial	This group is used for images that may change after several years. The default VisaPED script does not use this style group.

6.1 Style Example

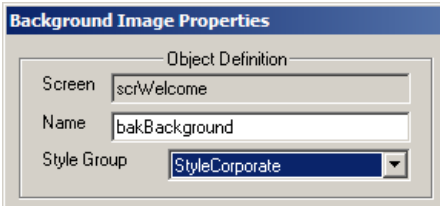
For our example, create a new script and add the default Welcome screen "scrWelcome" to the new script. This screen has a single background image called bakBackground that displays this image:

The default style group for the bakBackground control is "StyleLocked". This style group prevents styles from being applied to this control.



If you select "Project" + "Style" + "New" from the main menu you see the Style Wizard dialog pop up.

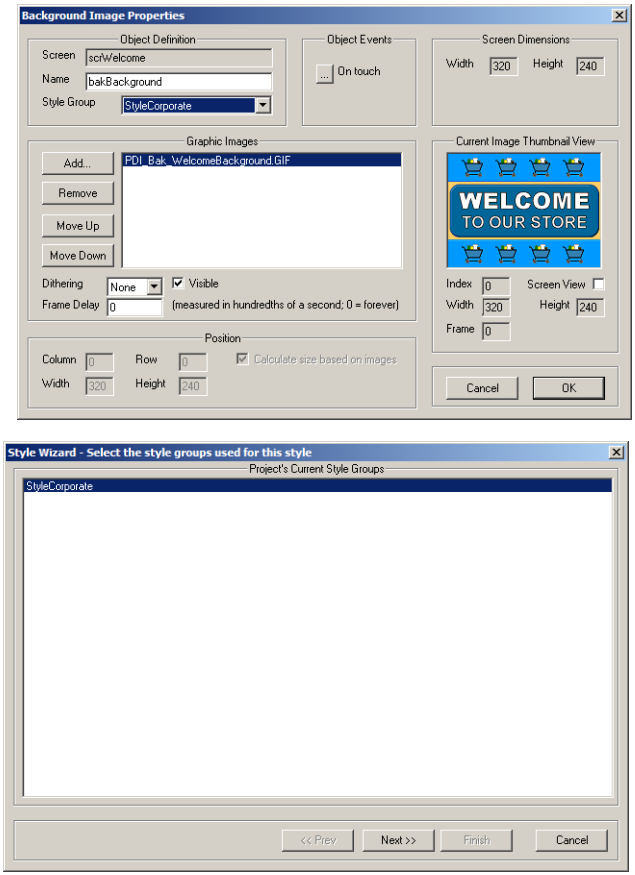
The "Project's Current Style Groups" section is empty, preventing you from create a style for the script.



For our example, select the bakBackground control, display its properties dialog and change the style group to “StyleCorporate”, then press “OK”. This will allow this control to be stylized.

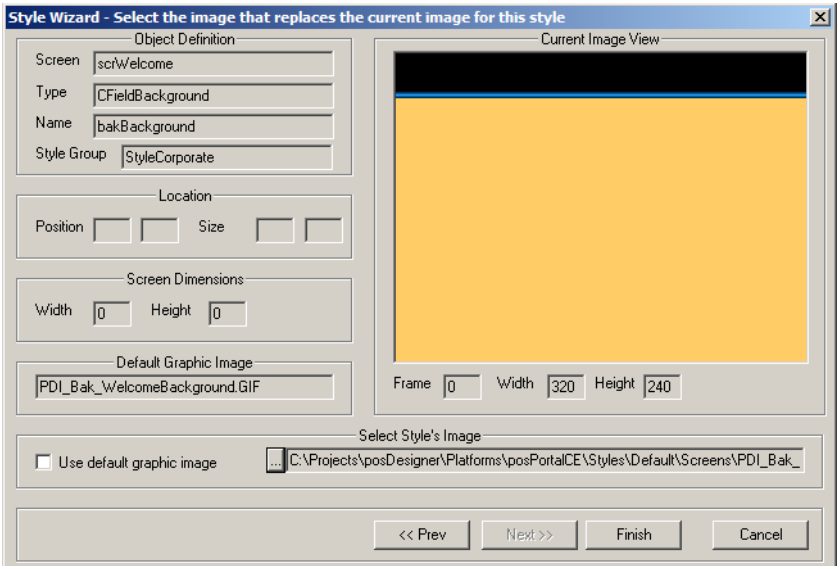
Select “Project” + ”Style” + ”New” from the main menu. The Style Wizard dialog pop up displays.

The “Project’s Current Style Groups” section contains the “StyleCorporate” entry . Select this style group and press “Next”.



The Style Wizard displays the next screen. Select a style image that replaces the image in the script.

By default, the "Use default graphic image" checkbox is selected. To create a style image to replace this image, select the browse button "…" next to the check box. An "Open File" dialog appears. Search for the new style image, select it and press "Open". For this example, select the file "PDI_Bak_BlankBackground.GIF" and press "Open".



The new file displays in the Style Wizard. Press "Finish".

A "Save New Script Style" dialog box appears. Type in the name "MyStyle.dss" and press "Save".

The script now displays the new style image in place of the actual image. The status bar in the lower right hand corner now displays "Style=MyStyle.dss" indicating the new style is in use.

To view the script using the original files, unload the style.

To unload the current style, select "Project" + "Style" + "Unload".

To load a preexisting style, select "Project" + "Style" + "Load".

Once a style is loaded, it can be edited using "Project" + "Style" + "Edit".

To save a loaded style after editing, use "Project" + "Style" + "Save".

To export the style to an IFB file, select "Project" +"Style" + "Export IFB". This allows you to upload the style to a POS terminal at a later time.

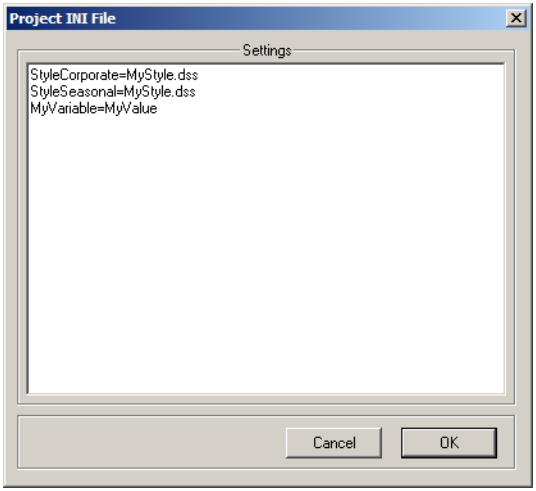
To upload the style to a POS terminal, use "Project" + "Style" + "Upload".

7

Project INI Files

posDesigner allows the developer to create INI files to apply to preexisting scripts. An “INI” file is a text file containing varilables and value pairs. The “INI” can be created and uploaded to a POS terminal to set new default values for a script.

To edit a project INI file, select “Project” + “INI File” + “Edit”. This brings up the “Project INI File” dialog box. Manually type in new settings for the POS terminal. Settings are typed as variable and value pairs. The variable name is separated from the value with the equal (“=”) character.



For example

```
StyleCorporate=MyStyle.dss
StyleSeasonal=MyStyle.dss
MyVariable=MyValue
```

In the above example, the variable “MyVariable” is set to the value “MyValue”. If uploading this INI file to the POS terminal using “Project” + “INI File” + “Upload”, it contains the new variables and their values when the POS terminal reboots.

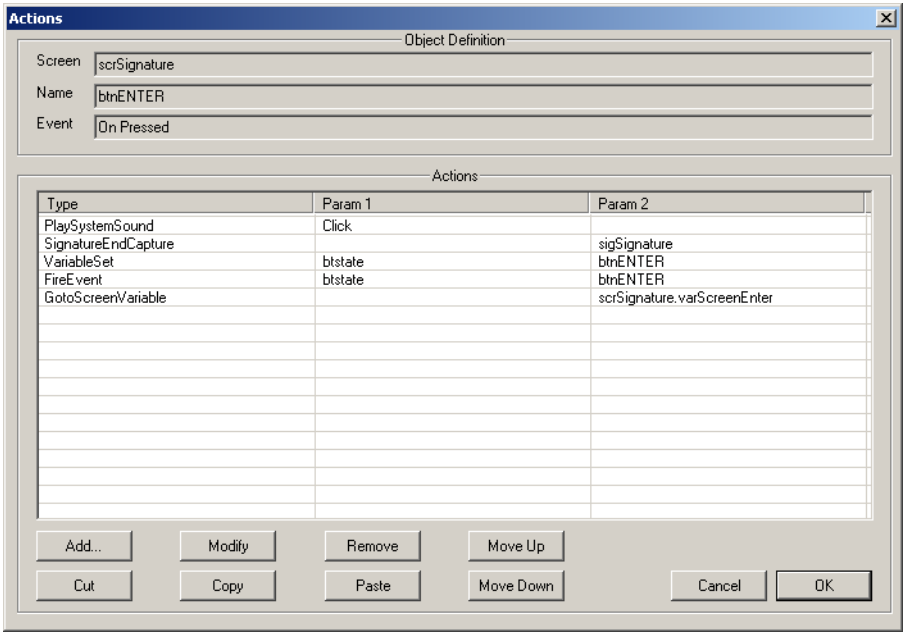
Any variable set from the cash register (host PC) can be set in the INI file. There are no restrictions.

INI files do not need to be digitally signed.

8

posDesigner Actions

Numerous posDesigner controls contain events resulting in actions when run on the POS terminal. A few of these actions (such as **GotoScreen**) are simulated in the emulator; most are not. Actions enable the developer to set variables, jump from one screen to the next and save signature data.



Controls supporting actions (such as the button control) have an **Object Events** section in their property dialog box. For example, Button Properties shows a button control has an On button pressed event. Pressing the **On button pressed** button displays the **Actions** dialog. This button (in the example) has two actions when pressed (PlaySystemSound, and GotoScreen).

Use the Actions dialog box to add, modify, remove, and re-order actions. To add an action, press **Add…** or double-click a blank space in the list of actions. This displays the Add Action dialog box. Select the action Type and its parameters. Some actions take label and data parameters while others only take one or the other. If an action only has one parameter, the other parameter field is disabled.

To modify an action in the Actions dialog box, press **Modify** or double-click the action to modify. This displays the Modify Action dialog. The Modify Action dialog has the same functionality as the Add Action dialog box.

8.1 ActionDelay

ActionDelay introduces a delay for a specified number of milliseconds. Set the data parameter to the number of milliseconds. ActionDelay has no label parameter.

8.2 BeginTransaction

BeginTransaction marks the beginning of a transaction. This is done by setting the following variables:

```
btstate= "0"
pinstatus= "0"
swipestatus = "0"
seskey= ""
sigcomp= ""
cindicator= "$"
```

BeginTransaction takes no parameters.

8.3 ClearTransaction

ClearTransaction clears the current transaction data. ClearTransaction takes no parameters. The following variables are considered part of the transaction and are cleared:

```
ttype
paytype
cnum
alltrack/txt/1/2/3
expd
fname
lname
pinblock
debitac
cashback
dtotal
sig
sigraw
date
```

```
total
stime
etime
litype, desc, price, sku, brand, subtotal, tax (Line item data)
```

8.4 ControlVisibilitySet

ControlVisibilitySet sets the Visible property of a control. Set the label parameter of ControlVisibilitySet to contain the name of the control to set the visibility of. Set the data parameter of ControlVisibilitySet to 1 to make the object visible and 0 to make the object invisible. Once set, the control display is updated.

8.5 DeviceReset

DeviceReset resets the POS terminal. DeviceReset takes no parameters.

8.6 DownloadMode

DownloadMode puts the POS terminal into download mode. DownloadMode is for future implementations. DownloadMode takes no parameters.

8.7 EndTransaction

EndTransaction ends a transaction. EndTransaction takes no parameters.

8.8 FireEvent

FireEvent sends a text **SetVAR** command event to the host ECR. Set the label parameter to contain the variable name and the data parameter to contain the variable value. The parameters can be any text data. The host ECR is free to interpret the **FireSetVAR** parameters in any way. It is recommended however that Symbol standards be followed.

Action	***Label parameter***	***Data parameter***
FireSetVAR	screen.object	action type

On the **scrLineItems** screen there is a button called btnScrollUp which plays a button click sound, then scrolls the **lstLineItems** Line Display control up one line. Add a **FireSetVAR** action to the button to send a **SetVAR** command to the host ECR to notify the line items were scrolled up once when the button was pressed.

Action	***Label parameter***	***Data parameter***
FireSetVAR	scrLineItems.btnScrollUp	On button pressed

Fire an event to the host ECR to tell the host ECR an **On swipe good** action has taken place.

Action	***Label parameter***	***Data parameter***
FireSetVAR	scrMsrCredit.msrCredit	On swipe good

The host ECR can know exactly which control and screen the SetVAR event came from and what caused the event.to be sent.

8.9 GotoScreen

GotoScreen jumps to the specified target screen. If the screen name doesn't exist, the action does nothing.

8.10 GotoScreenVariable

GotoScreenVariable jumps to a new screen based on the contents of the specified variable. If using GotoScreenVariable with the variable name ScreenNext given and the variable ScreenNext contains scrWelcome, the action jumps to the scrWelcome screen. If the screen name doesn't exist, the action does nothing.

8.11 InputFieldAppend

InputFieldAppend allows text to be appended to the specified Input Field control. Set the text to append parameter of this action to contain the text data to append to the Input Field control specified. For more information, refer to *Input Field Control* on page 5-29.

8.12 InputFieldAppendVariable

InputFieldAppendVariable allows the text contents of a variable to be appended to the specified Input Field control. Set the variable to append the name of the variable containing the text data to append to the Input Field control. For more information, refer to *Input Field Control* on page 5-29.

8.13 InputFieldClear

InputFieldClear clears the data captured in the specified Input Field control. For more information, refer to *Input Field Control* on page 5-29.

8.14 InputFieldSave

InputFieldSave saves the captured contents of the specified Input Field control. Set the Destination parameter to contain the name of the variable to save the contents to. For more information, refer to *Input Field Control* on page 5-29.

8.15 LineDisplayClear

LineDisplayClear clears the text of the current Line Display control on the screen. Set the Control Name parameter to the Line Display control being referenced.

8.16 LineDisplayDataAdd

LineDisplayDataAdd allows text to be appended to the specified type of Line Display data storage. Set the Text To Add parameter to contain the text to append to the Line Display data store. Set the Data Type ID parameter to the type of Line Display data storage being referenced. Any Line Display control displaying the specified data type shows the new contents

8.17 LineDisplayRefresh

LineDisplayRefresh refreshes the contents of a Line Display control. Set the data parameter to the Line Display control being referenced.

8.18 LineDisplayScrollDown

LineDisplayScrollDown scrolls down the text contents of the specified Line Display control on the screen. Set the Number of lines to scroll parameter to contain the number of lines to scroll. Set the control name parameter to the Line Display control being referenced.

8.19 LineDisplayScrollUp

LineDisplayScrollUp scrolls up the text contents of the specified Line Display control on the screen. Set the Number of lines to scroll parameter to contain the number of lines to scroll. Set the control name parameter to the Line Display control being referenced

8.20 PlaySound

PlaySound plays a sound frequency for a specified duration. Set the label parameter of this action to contain the name of a musical note you wish to play (for example "C1" plays the first octave "C" note) or the numeric value of any frequency. Set the data parameter of this action to contain total the duration the sound will play. This duration is measure in milliseconds. PlaySound only works on the iPOS TX.C platform.

Note	***Freq.***	***Note***	***Freq.***	***Note***	***Freq.***	***Note***	***PlaySound***
C1	131	C2	262	C3	523	C4	1047
C1#	139	C2#	278	C3#	555	C4#	1111
D1	147	D2	294	D3	587	D4	1175
D1#	156	D2#	312	D3#	623	D4#	1247
E1	165	E2	330	E3	659	E4	1319
F1	175	F2	349	F3	698	F4	1397
F1#	186	F2#	371	F3#	741	F4#	1483
G1	196	G2	392	G3	784	G4	1568
G1#	208	G2#	416	G3#	832	G4#	1664
A1	220	A2	440	A3	880	A4	1760
A1#	234	A2#	467	A3#	934	A4#	1868

Note	Freq.	Note	Freq.	Note	Freq.	Note	PlaySound
B1	247	B2	494	B3	988	B4	1976

8.21 PlaySystemSound

PlaySystemSound plays a predefined system sound (such as a bell, alarm, etc.). Select the type of sound to play from the System Sound drop down menu.

8.22 SignatureClear

SignatureClear clears the contents of the specified Signature control. This action does not clear any stored signature data; it only empties the contents of the display in the Signature control.. For more information, refer to SignatureEndCapture.

8.23 SignatureEndCapture

SignatureEndCapture saves the captured signature data in the specified control on the current screen. The data is saved to the standard “sig”, “sigRaw”, “RawData”, and “PointArray” variables. To save the signature data to another variable, use the Vasriable Copy action after using the Enter action to copy the signature data to another variable.

8.24 TextLabelAppend

TextLabelAppend allows text to be appended to a Text Input Label control. Set the label parameter of InputFieldAppend to contain the name of the Label control. Set the data parameter of InputFieldAppend to contain the text data to append to the Label control specified.

8.25 TextLabelSave

TextLabelSave saves the contents of a Static Text Label control. Set the Label name parameter to contain the name of the Static Text Label control to save the contents of. Set the Destination variable parameter of to contain the name of the variable to save the contents to. For more information, refer to *Static Text Label Control* on page 5-27.

8.26 TextLabelSet

TextLabelSet sets the contents of a Static Text Label control to a text string. Set the Label name parameter to contain the name of the Static Text Label control to set the contents of. Set the New Label Text parameter to contain the new string value display in the Static Text Label. Once set, the Static Text Label's display is updated.

8.27 TouchPadOFF

TouchPadOFF disables touch screen activity (currently only supported on the iPOS TX.C POS terminal). It takes no parameters.

8.28 TouchPadON

TouchPadON enables touch screen activity (currently only supported on the iPOS TX.C POS terminal). It takes no parameters.

8.29 VariableClear

VariableClear clears the data in a variable. Set the label parameter to contain the name of the variable to clear. If the variable name specified is a Label control, the Label is cleared and its display is updated. This action has no data parameter.

8.30 VariableCopy

VariableCopy copies the contents of one variable to another. Set the label parameter to contain the name of the variable to copy the data from. Set the data parameter to contain the name of the variable to copy the data to.

8.31 VariableSet

VariableSet sets the contents of a variable to a text string. Set the label parameter to contain the name of the variable to set the contents of. Set the data parameter to contain the new string value to set the contents to. If the variable specified does not exist it is created. If the variable specified is a Label control, its Label Data is set and the Label display is updated.

Symbol Customer Support

Symbol Technologies provides its customers with prompt and accurate customer support. Use the Symbol Support Center as the primary contact for any technical problem, question or support issue involving Symbol products.

If the Symbol Customer Support specialists cannot solve a problem, access to all technical disciplines within Symbol becomes available for further assistance and support. Symbol Customer Support responds to calls by email, telephone or fax within the time limits set forth in individual contractual agreements.

When contacting Symbol Customer Support, please provide the following information:

- serial number of unit
- model number or product name
- software type and version number.

North American Contacts

Inside North America, contact Symbol by:

- Symbol Technologies, Inc.
  One Symbol Plaza Holtsville, New York 11742-1300
  Telephone: 1-631-738-2400/1-800-SCAN 234
  Fax: 1-631-738-5990
- Symbol Support Center (for warranty and service information):
  - telephone: 1-800-653-5350
  - fax: (631) 563-5410
  - Email: support@symbol.com

International Contacts

Outside North America, contact Symbol by:

- Symbol Technologies
  Symbol Place
  Winnersh Triangle, Berkshire, RG41 5TP
  United Kingdom
  0800-328-2424 (Inside UK)
  +44 118 945 7529 (Outside UK)

Web Support Sites

MySymbolCare

http://www.symbol.com/services/msc

Symbol Services Homepage

http://symbol.com/services

Symbol Software Updates

http://symbol.com/service/downloads

Symbol Developer Program

http://software.symbol.com/devzone

Additional Information

Obtain additional information by contacting Symbol at:

- 1-800-722-6234, inside North America
- +1-631-738-5200, in/outside North America
- *http://www.symbol.com/*

Symbol Technologies, Inc.
One Symbol Plaza
Holtsville, New York 11742-1300
http://www.symbol.com

72E-64641-02
Revision A July 2004